MASHUP: A NEW WAY OF PROVIDING WEB MAPPING/GIS SERVICES

S. Li *, J. Gong

Dept of Civil Engineering, Ryerson University, 350 Victoria St., Toronto, Ontario, Canada M5B 2K3 –
(snli, jgong)@ryerson.ca

Commission IV, WG IV/5

KEY WORDS: Mashup, Internet GIS, System Integration, E-Government, Web Based, Services, Open Standards

ABSTRACT:

Recent development of Google Map, Yahoo! Map and MapQuest APIs have pushed map mashups almost to the top among other types of mashups such as news mashups, search and shopping mashups, and video and photo mashups. However, map mashups should not reply on these open data sources and services only. Developing a mashup in general faces a number of challenges. These include: 1) the difficulties in discovering and integrating legacy and local data into new mashup applications; 2) acceptable use policies such as the "protection of intellectual property and consumer privacy versus fair-use and the free flow of information” (Merrill, 2006); 3) sound security measures in mashups; and 4) for map mashups, the integration of map data sources and services with other types of data sources and services. This paper examines current development in mashup technologies, such as Google Map, Yahoo! Map, and MapQuest in the context of map mashups, and presents a classification of map mashups. Both technical and social issues related to the map mashup development and uses are discussed, with examples of case study mashups. The results demonstrate that map mashup has great potential to improve and facilitate the rapid development of the future web mapping/GIS services, such as participatory GIS and e-government services. However, issues related to data ownership, privacy protection, and some limitations inherent to the current mashup technologies need to be solved before its wider adoption.

1. INTRODUCTION

The term mashup is historically rooted in musical style compositions where recordings from different sources are combined into a new piece. On the Web it means taking information published from multiple sources and integrating it into a new information stream. A mashup therefore may be defined as a website or web application that uses contents or services from more than one source to create a completely new service (Wikipedia, 2007a). Mashup, in the form of hybrid web application, is a technology very often labelled under Web 2.0. As more and more studies being conducted towards integrating mashups into enterprise computing and e-government service environments, we will see more divisive mashup applications.

Arguably, Web 2.0 presents a second generation of web-based communities, applications and hosted services, which do not simply refer to an update to any technical specifications, but to changes of the ways software developers and end users use the Web (Wikipedia, 2008b). It promotes: 1) creativity, collaboration, and sharing between users (collective intelligence); 2) the move to the Internet and the Web as platforms (O'Reilly, 2005); 3) mass social networking channels; and 4) user generated and distributed web contents. The characteristics of Web 2.0 technologies lie in their ability to foster innovations and "assembling" systems, services, and websites by integrating features, services and data from distributed and, very often, independent developers. According to O'Reilly (2005), transforming from Web1.0 to Web 2.0 is a development from mere “read” to “write” and “build together”, which represents a revolution from the core content to external application. Mashup not only enables the traditional “read”, but also greatly fosters the “write” and “build together” by many.

Neogeography, a closely-related concept coined by Di-Ann Eisnor, co-founder of Platial, Inc., represents a new branch of geography that may consist of practices, theories, techniques and software tools that are different from the way the traditional GIS or professional cartographers handle maps and geo-spatial data. The way has simply been transformed into mashing up various web resources to create maps (Van Wyngaarden and Waters 2007). The term has emerged partly as a result of the vast amount of public interest in online mapping and using geo-spatial technologies, mostly through mashing up online map or map applications.

Accompanying with the emerging of these new concepts, recent development of Google Map, Yahoo! Map, MapQuest APIs, and others have pushed map mashups almost to the top among other types of mashups such as news mashups, search and shopping mashups, and video and photo mashups. One of the key drivers is the free map data sources and open APIs provided by these mainstream IT monsters. However, map mashups should not reply on these open data sources and services only. New ways of mashing up data and data services held by private companies and government agencies, together with other application-specific data and services, are needed.

Developing a mashup in general faces a number of challenges, both from a technical perspective and a social/organizational perspective. These include: 1) the difficulties in discovering and integrating legacy and local data into new mashup applications; 2) acceptable use policies such as the "protection of intellectual property and consumer privacy versus fair-use and the free flow of information” (Merrill, 2006); 3) sound security measures in mashups; and 4) the integration of map data sources and

* Corresponding author.

services with other types of data sources and services. Since map mashups have been widely developed for information sharing, user-added content integration, and social networking purposes in several categories, they are facing the same challenges with no exception.

This paper examines current development in mashup technologies, such as Google Map, Yahoo! Map, and MapQuest in the context of map mashups, and presents a classification of map mashups. Both technical and social issues related to the map mashup development and uses are discussed. Two experimental mashups for supporting e-government service are introduced and discussed to present a close examination of how geospatial data and services are integrated with others and managed in map mashups, map mashup architecture, development, acceptable user policies, and potential impact on future web mapping and participatory GIS as well as e-government services.

2. OVERVIEW OF MAP MASHUPS

This section serves as a brief introduction of map mashup which is the focus of this paper, and presents some basic facts, a brief review of key technologies, some design and development considerations, and current applications of map mashups.

2.1 Facts about Map Mashup

As stated by Ramsey (2006), "most mashups make use of a base map, a geocoder, and a Web interface." A map mashup therefore combines at least one map data source or service with added information, often geo-referenced to the map data, to create a new map (Pietroniro and Fichter 2007). One of the most popular mashup examples, *HousingMaps.com*, is actually a map mashup combining data and services from Google Map and craigslist – a data source for local classifieds and forums for 567 cities worldwide.

Surveying the total map mashups developed is not a trivial task. Figure 1 shows the statistics of various mashups created up to 27 April 2008 (completed by *programmableweb.com*), which include the top 10 categories of mashup websites. As shown in the figure, the map-based mashup accounts for a large proportion (39%) of the all registered mashups (the actual number at the time is 1618). This does not include other types of mashups that are not tagged as "mapping". With the relatively less or little requirements for programming skills and the increased number of mashup editors and creation tools, creating map mashups will become easier, resulting in more to come.

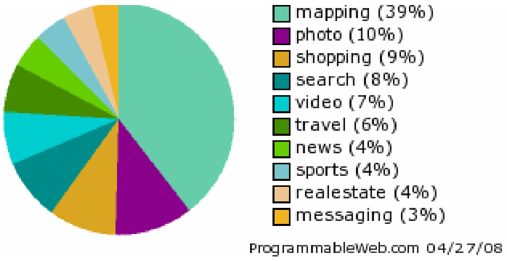


Figure 1. Mashup statistics by the Programmable.com

As mentioned in Section 1, Google Map, Yahoo! Map, MapQuest APIs, and others all have contributed to the fast growing of map mashups. However, it is the Google Map and Map APIs that may contribute the most, according to the statistics shown in Figure 2. Other significant contributors include APIs from Flickr and YouTube, which are primarily used to mash up image and video content with maps. For example, Flickr released two new map-related features at the end of 2007: Flickr Places and maps page. The Places allows users to drag and drop photos directly onto a map location, while the maps page provides a world view where people can easily view popular and up-to-date themes.

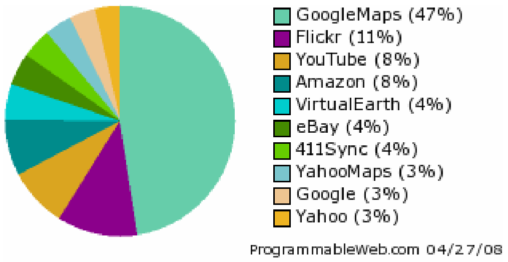


Figure 2. Statistics on map APIs used for map mashup by Programmable.com

While the majority of map mashups are simply for users to add locations or location-related data content onto the map (i.e., geo-referencing data content), we have also seen many mashups with more complex interfaces and powerful functions. For example, mashing up data services from Google Maps, Flickr, YouTube, and a real estate listing to create a comprehensive real estate website is now possible. In addition, more and more location intelligence has been built into mashups. A good example is the recent feature from Yahoo, called Yahoo! Shortcuts. With a plug-in, it allows automatic detection of location-related words or phrases such as places, companies, products, and people's names, and embeds hyperlink to the corresponding maps.

2.2 Related Technologies

A key technology for mashup is the Web APIs used to interface to various data and data services required by it. Up to this writing, *Programmableweb.com* has listed 737 APIs, and among them 55 APIs are mapping related. While some of these APIs are free and open for the public use with the conditions on rate limits, others are proprietary in nature and require fees for access and use. Nevertheless, these web APIs retrieve content from websites, which provide data and data services.

There are two camps of Web APIs in the context of map mashup: map APIs and non-map APIs. Map APIs will be further explained in Section 3.1. Non-map APIs are the ones that interface to the code that provides mashup access to other types of data and data services such as photos, videos, news texts, and real estate listings. Both Flickr and YouTube APIs present widely-used examples in this camp.

Other technologies that are important to mashups (including map mashups) are: Asynchronous JavaScript and XML (Ajax), Really Simple Syndication (RSS) or GeoRSS, Representational State Transfer (REST), services-oriented access protocol (SOAP), web services, semantics web, screen scraping, etc. (Merrill, 2006;). The following briefly summarizes some of these related technologies.

AJAX: creates interactive web applications, which allows the change of the part of web page content without reloading the display and behaviours of the page. This is realized through asynchronously requesting extra data from the server and loading it into the web page display in background. Merrill (2007) argued that Ajax is more a web application model, rather than being a technology. Compared with the traditional web application model, a web application based on the Ajax model eliminates the traditional start-stop-start-stop nature of interaction by introducing an intermediary, i.e., an Ajax engine, between the user and the server (Garrett, 2005). The Ajax engine is responsible for both rendering the interface and communicating with the server on the user's behalf at the same time. Ajax technologies have been widely used in web applications to provide such functions or usages as form data validation, auto-completion, page loading balance, partial data submission, sophisticated user interfaces, etc.

RSS/GeoRSS: RSS is a text-based web content syndication format, often specified using XML. Files related to RSS are normally called RSS files, or RSS feeds or channels. In order to read a RSS feed, software called "RSS reader", "feed reader", or "RSS aggregator" is required. There are two ways in reading RSS: subscribing and entering the RSS feed's link into the RSS reader or using the RSS icon embedded in a browser to start the subscription process. RSS can be used for syndicating not just the news and content of news-like sites, but also anything that can be broken down into discrete items (Pilgrim, 2002). GeoRSS is an emerging standard for RSS feeds to be described by location or geo-tagged in a standardized way, in which the location is encoded (Reed 2006). Many map APIs support the GeoRSS feeds with either coordinate (lat/long) data or address information specified in XML items. The later option requires a geocoder.

REST: a technique of Web-based communication using just HTTP and XML. REST supports only a few operations (i.e., POST, GET, PUT, and DELETE) that are applicable to all pieces of information, called resources (Merrill, 2006).

SOAP: a protocol for exchanging XML-based messages between Internet/Web applications over computer networks (Wikipedia, 2008c). Over the Internet, a SOAP message is transmitted through HTTP protocol, and also can be transmitted by other ways (such as JMS or E-mail). Based on SOAP specifications, APIs can be developed for interacting with web mapping services.

Web Service: a self-contained, self-describing, modular software component or application accessible over a network (e.g., Internet) based on its service description expressed in web service description language (WSDL). Web services must be loosely coupled, location transparent, and protocol independent (Endrei at al., 2004). The service consumer has no need to know the implementation details of the service. Using web services, the web is used for delivering not just geo-spatial data, but geo-processing functionality that can be created or wrapped in interoperable software components (Anderson and Moreno-Sanchez, 2003).

Technologies such as REST, Web Services, and RSS/GeoRSS are all Web-protocols facilitating retrieval of data content provided by data and service providers. For those data content and services without appropriate APIs, there are two ways for mashups to retrieve their data content: using screen scraping (Merrill, 2006) or using adaptors – software wrappers that adapt the existing APIs to the ones accepted by mashups (Huang and Chuang, 2006; Wikipedia, 2008d).

2.3 Map Mashup Classification

Currently there is no widely-accepted classification of mashups, including map mashups. Wikipedia (2008a) states three types of mashups in general: consumer mashups, data mashups, and business mashups, which may fall in four main categories, mapping, video/photo, search/shopping and news. Hinchcliffe (2006) presents a five style of mashups in terms of where mashup software may be placed on multi-level application stack. These styles include presentation mashup, client-side data mashup, client-side software mashup, server-side software mashup, and server-side data mashup. In addition, different map mashups have been named based their technology integration, e.g., WikiGIS and MapBlog.

The difficulties in classifying all mashups lie in the facts that: 1) the overwhelming number of map mashups available on the Web; 2) the difficulties of drawing clear boundaries between different map mashups; 3) understanding of what is being mashed up, i.e., data, visual presentation or underlying functionality (code) (Hinchcliffe, 2006); and 4) a large number of new technologies involved in developing mashups.

An easy way to classify map mashups is to group them from user and usage perspective into four categories: informative, participatory, collaborative, and enterprise. The informative mashups have the lowest complexity level, while the enterprise mashups have the highest. Accordingly, the requirements for programming skills are increasing as the level moves up. For example, the mashups created using the lightweight tools, as discussed in Section 3.2, will fall into informative category. An enterprise mashup could be any of these types. A major feature distinct enterprise mashups from the others is that enterprise mashups mashes data content not only from the Internet but also enterprise information systems.

Informative: designed and created for information presentation and browsing, not content creation, e.g., Chicago Crime Map, Housing Maps, and Mappr; mashups in this category deal with one-way information flow, and may also include those sometimes labelled as "passive participation" (Huang and Chuang, 2006).

Participatory: Focuses on "initiative participation" where people can add data content, initiate a participatory process, and interact with the content other people have added.

Collaborative: in addition to the two-way participation, it also allows sharing of the data content and collaboration between users using maps as a platform. The collaboration can be either asynchronous or synchronous by remixing a real-time chat service, for example.

2.4 Applications

Map mashups are currently capable of: 1) embedding maps with simple GIS/mapping functions (e.g., pan and zoom) into various web pages or sites; 2) annotating or marking up other location-related content on maps; 3) providing location intelligence; and 4) supporting data content search and query.

A search on the Internet and try-out of some map mashups tagged as "mapping" on *programmableweb.com* website show that most map mashups are created for applications in real estate (e.g., resale home search), business locations (e.g., beer stores), transportation and transit mapping (e.g., metro transit map), news incident locations, social networking (e.g., buddy mapping and sharing photos and videos), tourism, crime mapping, simple community services such as event mapping, etc. Table 1 lists some of the websites that provides links to example mashups for different types of applications. Few has been found in providing map-based government services and formal participatory decision making which involve two way communications flow, or mashing up the data sources and services provided by government agencies so that the citizens can carry out their own analysis using lightweight mashups.

Site Name	URL (web address)
ProgrammableWeb.com	http://www.programmableweb.com/tag/mapping
MashupCamp	http://wiki.mashupcamp.com/index.php/BestMashupContest6
Google Demo Gallery	http://code.google.com/apis/maps/documentation/demogallery.html
Google Maps Mania	http://googlemapsmania.blogspot.com/
Yahoo Map Gallery	http://gallery.yahoo.com/maps
OpenLayer	http://www.openlayers.org/dev/examples/example-list.html

Table 1. Websites linking to mashup applications

3. MAP MASHUP DEVELOPMENT

According to “How to Make Your Own Web Mashup” on the *Programmableweb.com* website, creating a mashup only involves a five step process: pick a subject, decide your data sources, weigh your coding skills, sign up for an API, and start coding. MashupAwards site (*http://mashupawards.com/*) has outlined a few important things in general which should be considered before starting to build a mashup. These include: coming up with a simple, useful or practical idea, choosing the right tools and right data sources, and deciding how to host your mashup. Sound easy? Not really! Mashups range from simple GUI interface to more programming straight-on (Pietroniro and Fichter 2007). Developing a good mashup, especially the one for enterprise and government services, requires:

- Better understanding of programming skills involved;
- Sufficient knowledge of the involved technologies and tools;
- Clear definition of the intended purposes and requirements based on intended business processes; and
- Good understanding of potential social/organizational implications, as well as technology restrictions.

3.1 Map APIs

Map APIs are currently provided by mainstream IT firms such as Google and Yahoo, proprietary GIS software vendors (e.g., ESRI ArcWeb API), government organizations (e.g., UK Ordnance Survey), and in some cases, research projects. Many of them are available for free to the public provided the use of these APIs is not for commercial and/or production purposes. Some API providers have set up the limits restricting the number of transactions to their data severs through the developed mashups. Most APIs are JavaScript-based, taking advantage of AJAX technology.

While we have no intention to present an in-depth technical review of all available map APIs, the following lists some major ones with brief information about them and the sources for more information:

Google Map API: free beta service with rate and usage limits (http://code.google.com/apis/maps/index.html)

Mapstraction: free access to codes, a JavaScript library that provides a common API for various JavaScript mapping APIs to enable switching from one to another (www.mapstraction.com)
MapQuest API: no fees for building applications for publicly available websites, provides JavaScript, Flash/Flex, Java, C++, and .NET APIs (http://developer.mapquest.com/)

Microsoft Live Dev API: free with rate and usage limits (http://dev.live.com/virtualearth/)

Multimap OpenAPI: provides free access to static, Ajax and web service APIs with transaction limit, data and data services have an European flavour (www.multimap.com/openapi/)

OpenLayers: free with open source code (www.openlayers.org/)

Yahoo! Map API: free with rate and usage limits, provides simple, flash and map image APIs in addition to JavaScript API (http://developer.yahoo.com/maps/)

Selecting the right API is important for developing map mashups. However, a quick search of literature finds little on evaluating different map APIs, tools or platforms. Proto (2007) describes a set of 17 criteria for evaluating mashup building platforms for business applications. Many of these criteria are closely related to programming level evaluation such as at widget level. For elevating map APIs, more focus should be put on its supported functionality, data integration options, data ownership, and technology compatibility, if multiple data services are to be mashed up.

3.2 Mashup Building/Editing Tools

Creating mashups often requires various levels of programming skills which are not in the reach of non-programming end users. To help end users and programmers to easily and quickly build mashups, several API providers have released sort of mashup building/editing tools. While some of these tools are dedicated to building simple map mashups, others are more general and can be used to create any mashups using the Web APIs.

There are a number of lightweight tools for non-programming end users to create their own simple map mashups that add location-based content. Typical examples include MapBuilder (*www.mapbuilder.net*), YourGMap (*www.yourgmap.com*), and Platial (*www.platial.com*). Once the users create their mashups, they can either obtain the source codes and insert them into the existing website or directly publish the maps to the tool provider’s site. However, these mashup building tools do not help create more complex map mashups.

Mashup editors, on the other hand, are interactive development environment (IDE) and require certain level of programming skills and knowledge to use. They are platforms or tools that help users (programmers) to create mashups more easily and efficiently. There are a number of mashup editing tools available to choose; examples include Yahoo! Pipes, Microsoft

Popfly, and Google Mashup Editor. At the time of this writing, Google Mashup Editor is still in testing stage and only the people who request and are authorized can use it.

While the details of the three mashup editors mentioned above may be found from the website listed below, a number of points should be noted here. First, all three tools follow the WYSIWYG principle by providing "preview" or "test" function that allow users to test their mashup. Second, only Yahoo! Pipes and Microsoft Popfly allow drag-and-drop style of editing to add data and data service components into the mashup interface. Third, all of them require registration to access, but the use is free. And finally, the mashups created using these tools may only be published to and hosted by the tool provider's server.

- Google Mashup Editor: http://code.google.com/gme/, aimed at developers familiar with HTML and JavaScript, who want to deploy Ajax user interface components atop existing feeds and Google web services
- Microsoft Popfly: http://www.popfly.com/
- Yahoo! Pipes: http://pipes.yahoo.com/pipes/

3.3 Architecture Design Considerations

Mashup typically follows 3-tier architecture following a mix of client/sever and web service model. As Merrill (2006) states, a general mashup architecture comprises of three different participants that are logically and physically disjoint. Figure 3 shows this generic architecture with mashup client sitting on the top and the data sources and services sitting on the bottom. The middle tier is where the mashup logics reside. It should be noted that the mashup logics for generating mashed content could be either executing on the server or within the web browser.

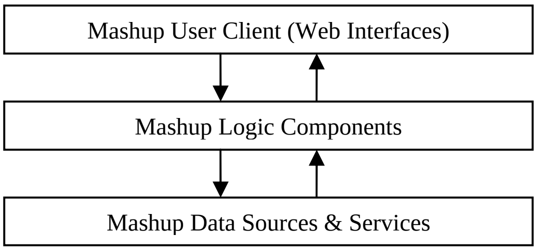

Mashup User Client (Web Interfaces)
Mashup Logic Components
Mashup Data Sources & Services

Figure 3. Generic mashup architecture

The architecture for enterprise mashups may consist of a traditional client and server framework, as well as a set of web services and JavaScript APIs. Since many business services in an enterprise used to be developed with Java, there is a potential incompatibility between Java and JavaScript. In other words, enterprise mashups need to deal with both JavaScript APIs and those JavaBeans-based services. Although some Web APIs such as MapQuest API and Flickr API support Java, the majority of other APIs do not. An interim solution, although cumbersome in processing XML with JavaScript, is to expose enterprise business services as web services so the SOA can play its role in enterprise mashups (Smith, 2007).

4. CASE STUDIES: MASHUPS FOR E-GOVERNMENT SERVICES AND PUBLIC PARTICIPATION

In order to study the potential issues and possibilities of mashing up applications for supporting e-government services, we have created two simple map mashups, which are only served for the purpose of case studies.

4.1 Mashup for West Nile Related Reporting

West Nile is normally spread by mosquitoes after becoming infected by feeding on the blood of infected birds according to Public Health Agency of Canada (www.phac-aspc.gc.ca/wn-no/index-eng.php). Reporting dead birds is the first step the public can help prevent West Nile virus in our community. The current reporting methods at three levels of governments in Canada are all based on paper form or text-based e-forms, in which the geographical location data needs to be filled. This case study tries to develop a map mashup which allows the public to report dead bird with more accurate location using a map interface, facilitated by other data entry interfaces.

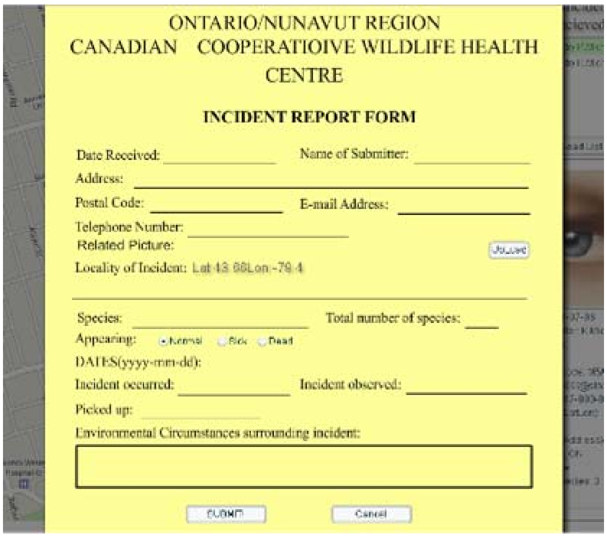
The form-based reporting of dead birds requires different types of data depending on the individual jurisdictions. Two important elements in the reporting are the location where the dead bird is found and the identification of the bird type. For the location information, most provinces require the closest address in city and GPS coordinates (latitude and longitude) in rural areas. To identify the bird type, the reporter needs to check the bird pictures and descriptions posted on related government agencies or organization's websites. It is not surprising that reporters often make the incorrect or inaccurate judgement because: 1) the dead bird may look out of its shape comparing to the ones in the pictures; and 2) the closet street address may be wrongly picked. As such, the required data to report the dead birds using the reporting map mashup should include:

- Dead bird finder's information
- Reporter's information, if different from the finder's one
- Dead bird location information, allowing both map pinpointing and direct input of civic address and/or coordinates
- Description of the reported bird and its surround environment, and photos, if taken by the reporters

The main purpose of using a map mashup is to facilitate the submitters to identify the location of the dead birds found. Once the location is identified, the corresponding coordinates are automatically filled in the form. If the submitter is preferred to enter the closed street address, its location will also be geocoded on the map. This will help create a complete database of the reported species. The further development of this mashup will also allow the public to browse the reported incidents (without the submitter's personal information) and view the related statistics.

Two sets of APIs and technologies were tested in creating such a mashup. The first one used Yahoo! Map API, Flickr API and flash technology, as show Figure 3. The second set used Google Flickr API and HTML forms, as shown in Figure 4.

The main problems encountered during the mashup prototyping are related to the technology incompatibility and data uploading and holding. For example, in creating the mashup shown in Figure 3, the web access address of the photos uploaded to Flickr cannot be directly picked up by the mashup due to the lack of support of Flickr APIs to interact with flash components. In addition, the photos are actually uploaded to the Flickr server instead of the mocked application server. In both mashups, we were able to upload the form data to our mocked application server through extra codes, which is basically a record in database table with coordinates of the reported location. This means when viewing the reported incidents, they will need to be mashed up with free map data again.



ONTARIO/NUNAVUT REGION
CANADIAN COOPERATIOIVE WILDLIFE HEALTH CENTRE

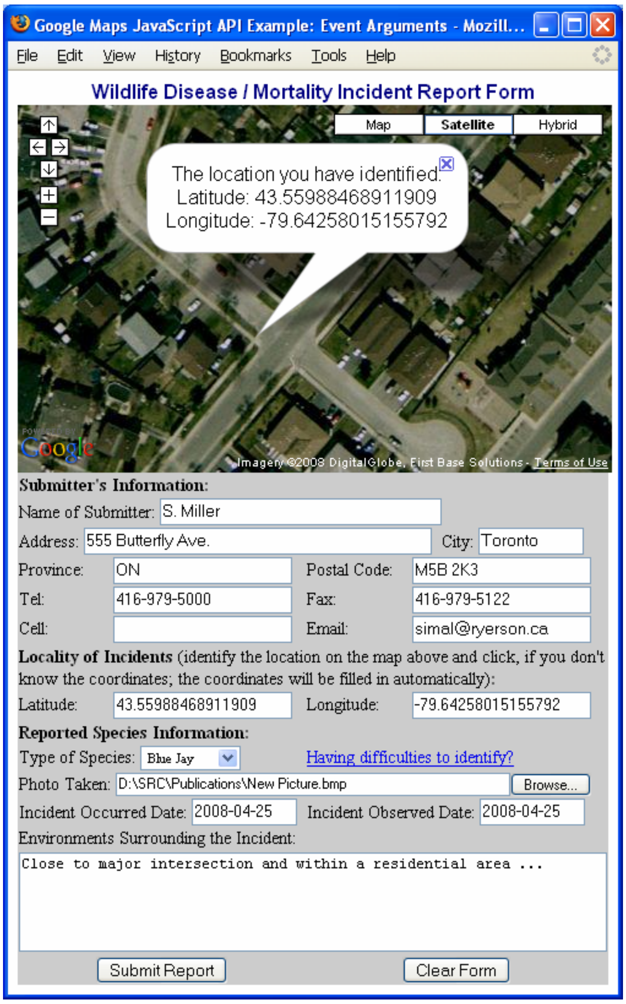
INCIDENT REPORT FORM

Date Received: ____ Name of Submitter: ____
Address: ____
Postal Code: ____ E-mail Address: ____
Telephone Number: ____
Related Picture: Upload
Locality of Incident: Lat 43.58Lon -79.4

Species: ____ Total number of species: ____
Appearing: ● Normal ○ Sick ○ Dead
DATES(yyyy-mm-dd):
Incident occurred: ____ Incident observed: ____
Picked up: ____
Environmental Circumstances surrounding incident:

SUBMIT Cancel

Figure 3. Dead bird reporting mashup using Yahoo API, Flickr and Flash


Google Maps JavaScript API Example: Event Arguments - Mozill...
File
Edit
View
History
Bookmarks
Tools
Help
Map
Satellite
Hybrid
POWERED BY Google
Imagery ©2008 DigitalGlobe, First Base Solutions - Terms of Use

Wildlife Disease / Mortality Incident Report Form

The location you have identified:
Latitude: 43.55988468911909
Longitude: -79.64258015155792

Submitter's Information:
Name of Submitter: S. Miller
Address: 555 Butterfly Ave. City: Toronto
Province: ON Postal Code: M5B 2K3
Tel: 416-979-5000 Fax: 416-979-5122
Cell: Email: simal@ryerson.ca

Locality of Incidents (identify the location on the map above and click, if you don't know the coordinates; the coordinates will be filled in automatically):
Latitude: 43.55988468911909 Longitude: -79.64258015155792

Reported Species Information:
Type of Species: Blue Jay Having difficulties to identify?
Photo Taken: D:\SRC\Publications\New Picture.bmp Browse...
Incident Occurred Date: 2008-04-25 Incident Observed Date: 2008-04-25
Environments Surrounding the Incident:

```
Close to major intersection and within a residential area ...
```

Submit Report Clear Form

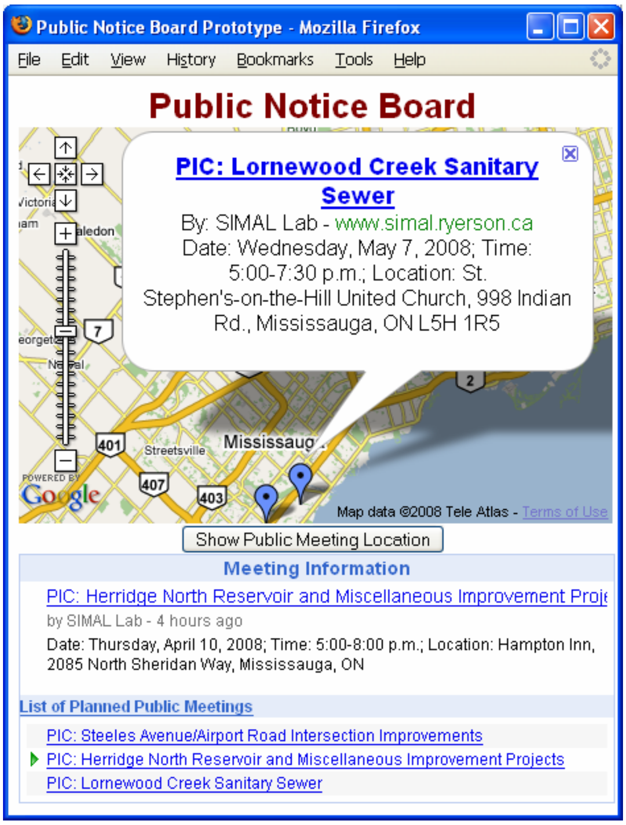
Figure 4. Dead bird reporting mashup using Google API, Flickr API and HTML form

4.2 Public Notification

In the context of municipal planning and development, public notification or notice serves the purpose of notifying the public about the upcoming public meetings, public information centres or status of environment assessment studies. According to Li, et al. (2007), the main role of the public notification is to notify and provide information about the project under public consultation, in addition to others.

The map mashup illustrated in Figure 5 has been created using Google Map API and GeoRSS technology. The public meeting events are mapped as "markers", which display the detailed information about the meeting in the information window. The list of public meetings come directly from a GeoRSS compatible feed. The HTML-based info window can be easily expanded to include functions such as driving direction from a location to the meeting place and link to another map mashup interface for exploring the project area possibly with map-based discussions.

For the exploration of project area and proposed development geometry, it is easy to use Google map's feature such as point, line and polygon markers to create a mashup interface. One problem in testing the mashup using current public notice data posted on the Internet is the coverage of open maps. In some cases, the proposed development is in new areas which may fall in those "no data areas". If more spatial information other than drawing simple points, lines and polygons is needed to facilitate the public to understand the proposed development, the data held by municipal planning department may need to be overlaid, which is not easy to mash up.



Public Notice Board

PIC: Lornewood Creek Sanitary Sewer
By: SIMAL Lab - www.simal.ryerson.ca
Date: Wednesday, May 7, 2008; Time: 5:00-7:30 p.m.; Location: St. Stephen's-on-the-Hill United Church, 998 Indian Rd., Mississauga, ON L5H 1R5

Show Public Meeting Location

Meeting Information

PIC: Herridge North Reservoir and Miscellaneous Improvement Proje
by SIMAL Lab - 4 hours ago
Date: Thursday, April 10, 2008; Time: 5:00-8:00 p.m.; Location: Hampton Inn, 2085 North Sheridan Way, Mississauga, ON

List of Planned Public Meetings

- PIC: Steeles Avenue/Airport Road Intersection Improvements
- PIC: Herridge North Reservoir and Miscellaneous Improvement Projects
- PIC: Lornewood Creek Sanitary Sewer

Figure 5. Public notice board mashup interface

5. TECHNICAL AND SOCIAL ISSUES

The challenges and issues discussed here are mainly based on our case studies presented in the previous section as well as the review of the corresponding literature. No means to be complete! We rather treat them for further discussion purpose.

5.1 Technical Challenges

Technical challenges in dealing with map mashups are mainly related to technical incompatibility, data integration and quality assurance, interfaces and functionality, level of programming skills required, enterprise integration, etc.

Dealing with many incompatible technologies and technical specifications, either associated with different technologies or with the same technology, causes a major challenge to map mashup development. For example, there are a number of formats of RSS feeds, including three main GeoRSS formats (namely W3C geo, Simple GeoRSS or Pro GeoRSS, see Table 2). Ensuring some level of compatibility is an issue to consider in terms of what format is chosen to form the feeds and how to handle all the variations when reading them (Shutzberg, 2006).

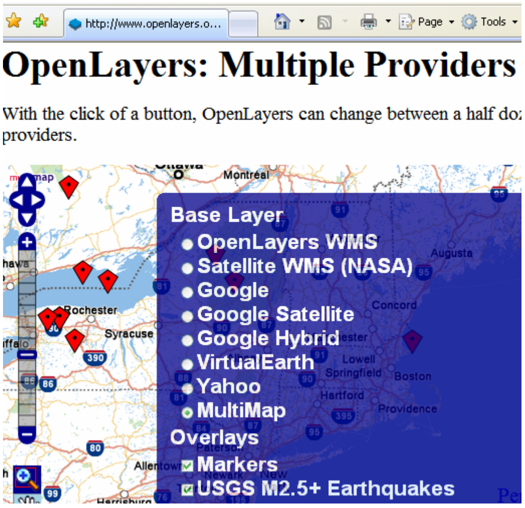
Format	Feature types supported	Ref. datum
W3C Geo	point	WGS84
Simple GeoRSS	point/line/polygon	WGS84
Pro GeoRSS	point/line/polygon	more

Table 2. Major GeoRSS formats

An article published by SYS-CON Media, Inc. states two problems concerning enterprise adoption of mashups: “tangential to enterprise software” and “no clear path to integrate business services” with available web services (SYS-CON, 2008). One of the key technical problems mentioned is the integration between JavaScript and Java.

Map APIs are often tied up with data and data services provided by the API providers. This raises questions such as: 1) how can we integrate the other data, especially legacy data, with the data freely provided; and 2) how to ensure the quality of data? Study carried out by Nivala et al. (2007) indicates that all free map services used in his study have various problems with data coverage, accuracy and currency, as well as metadata information. Typical problems found include browsing to a display showing “data not available”, coarse resolution after zooming into a certain level, mismatched data, and most importantly, no information about the quality of the data viewed. This has great implications for how the services should be used and what applications may be built upon these services.

Building more useful analytical functions into map mashups is more a question rather than challenges. The exiting open APIs provide overlay capability of adding user-added data, mostly geo-referenced “attributes”, as well as points, lines and polygons onto the maps. However, they don’t really provide the same level of support of spatial analysis as we use to enjoy from a simple desktop GIS. OpenLayers API allows overlay of data from different data sources (see Figure 3). Before we can see some real analytical functions, the Open APIs may need to provide interfaces to accessing its base data at feature level.



http://www.openlayers.o...

OpenLayers: Multiple Providers

With the click of a button, OpenLayers can change between a half do providers.

Figure 3. OpenLayers integrates data from multiple providers (http://www.openlayers.org/gallery/multiple.html)

The new interfaces provided by Google Maps, Yahoo! Maps and so on have greatly changed the users’ perception of web mapping/GIS. Maps play an important part for displaying and receiving geo-referenced, user-added content. However, lack of meaningful and quality cartographic design is still of concern from both professional cartographer and user’s perspectives. The problems existed in not only the base maps provided by service providers, but also the representations of user-added data, e.g., the use of default “marker” symbols for different things. In many cases, the symbols, colours, use of space, etc. are not clear, meaningful, and consistent, e.g., inconsistent presence of zooming tools and scale bars (Nivala et al. 2007).

While free maps and map APIs may be good for social mapping and providing general location-based information, they may be questioned if used for some mission critical applications such as emergency planning and response, as well as within enterprise computing environments. For the later, in addition to all of the above mentioned challenges, the extensive programming effort also needs to be considered.

5.2 Social and Organizational Issues

5.2.1 Privacy and Ownership

Mashups, or more generally Web 2.0 paradigm, are, in fact, highly user-centric, meaning that their focus is on enriching users' interactive online experiences by allowing them, not necessarily a professional programmer, to freely create their own things using others' networked resources without getting their prior permission. This raises the biggest social issue facing mashup developers, which is “the trade-off between the protection of intellectual property and consumer privacy versus fair-use and the free flow of information” (Merrill, 2006).

A number of questions are raised regarding the user-added content through mashups. Where are the data uploaded to? Who owns the data? Who is benefiting from this collective intelligence? What about the data privacy? A simple mashup may allow data to be uploaded to a number of servers where the services are provided. Although, the mashup can be designed in such a way that the user-added data may be diverted to the preferred server hosting it, there are always some data being uploaded (duplicate copy) to the API provider’s server due to its restrictions. Who hold the user-added data is concerned with not only data privacy and ownership, but also who can benefit from the user-added content, so-called "collective intelligence". In the world of mashups, data owners actually hold the balance of power (MacManus, 2008) and are the ones who really benefit from the collective intelligence (Walsh, 2008).

While service API providers normally have clear terms of use on data policies, privacy and ownership, the grey area is in those applications or mashups developed using the open APIs. Specially, when mashing up applications for supporting e-government services which data privacy is of major concern, it is important to have a better understanding of this issue.

5.2.2 Copyrights and IP Protection

Copyright matters become more complex when dealing with the combined content from various independent sources using mashup technologies. While all mashup API providers have outlined their policies on using their APIs and the associated map data services, it is the content combined from other sources that create potential problems. A number of studies have been reported on this issue, e.g., Taranto (2007) presents a court case in Australia which holds the designer and ISP provider responsible for a website providing links to copyrighted materials (MP3 files).

According to O'Brien and Fitzgerald (2006), "mashups and remix will inevitably encounter legal problems when the whole or a substantial part of the original material has been reproduced, copied, communicated, adapted or performed". In the case of map mashup, the original source data may not be reproduced and copied. Rather, part of them may be shared with the added content, aggregated and analyzed to offer new information. Are there any legal implications under these situations?

It is suggested that the current copyright laws be reformed to accommodate this mashup and remix phenomena, or at least as an interim solution, what's called "Creative Commons" be considered (O'Brien and Fitzgerald, 2006; Taranto, 2007). Creative Commons provides licenses and/or tools that allow copyright holders to grant some or all of their rights to the public for sharing, remixing and reuse (Wikipedia, 2008d).

5.2.3 Acceptable Use Policies

An important issue in developing map mashups using open APIs and map data services is related to their acceptable use policies which need to be examined more closely (Pietroniro and Fichter, 2007). Ramsey (2006) discusses about the "fine print" related to Google's right to potentially add ads on its base maps in the future and the requirements for the users to use its API only for mashups accessible for free. Some of important terms need to be considered if one plans to use the API to develop web mapping applications (Li, 2008). These include:

- *Rate limit*: controlled based on the number of access or queries to map data services through your application created using the API; for example, Yahoo has a varied rate limit depending on which set of APIs you use. Except for Yahoo! Maps Simple API, all other APIs currently have a rate limit of 50,000 queries per IP every 24 hours.
- *Advertising*: all vendors reserve the right to add ads in the map images provided through the API service, with or without notice prior to the commencement of doing so.
- *Charge of API use*: all vendors currently provide services for free with a condition that they may be used only for services that are generally accessible to consumers without charge. An access key or ID is required for map APIs.

Other itchy terms requiring attention of those who would like to develop real-time mapping mashups using the free map APIs are the restrictions on adding real-time tracking data obtained from any tracking systems and devices to the maps (Li, 2008). The level of restrictions varies from one vendor to another.

6. CONCLUSIONS AND FUTURE IMPACTS

Map mashups have great potential to improve and facilitate the rapid development of the future participatory GIS and e-government services. However, issues related to data ownership, privacy protection, and some limitations inherent to the current mashup technologies need to be solved before its wider adoption.

It is clear that the applications of map mashups in providing online, web-based government services and facilitating enterprise computing environment have not been well developed. The level and extent of APIs provided by government is low. The government agencies that hold geospatial data and provide data services should develop more open APIs so that their data services may be mashed up into new generation of web mapping/GIS services.

In the near future we will see the development of Web 2.0 and more particularly, map mashups in a number of directions:

Social Mapping and Networking: mapping user-added data content and social connections will continue to be an important area of map mashup.

Enterprise Computing: according to a report by Forrester Research, enterprise spending on Web 2.0 technologies will be growing 43% each year to reach $4.6 billion globally by 2013. The main driving force of this surge will be social networking and collaborative power that can be incorporated in the enterprise environments.

E-Government Services: two primary principles implied by Web 2.0 are participation and user-created valuable content. These are well in line with the current practices the governments provide their e-services. It is expected government agencies will follow the concept and develop more open APIs to its holding data so that they can be better utilized by others.

Map mashups and, in a more broad sense, Web 2.0 are more than merely social networking websites. Serious studies and investigations are required to see how best we can take advantages of this new collection of concepts, tools and technologies to develop web GIS/mapping services, especially in the area of participatory and collaborative map mashups for communities and e-government services, as well as enterprise information management.

ACKNOWLEDGEMENT

The work involved in this paper has been funded by a Discovery Grant from the National Science and Engineering Research Council (NSERC) of Canada.

REFERENCES

Anderson, G. & Moreno-Sanchez, R., 2003. Building web-based spatial information solutions around open specifications and open source software. Transactions in GIS 7(4): 447–466.

Endrei, M., Ang, J., Arsanjani, A., Chua, S., Comte, P., Krogdahl, P., Luo, M. & Newling, T., 2004. Patterns: service-oriented architecture and web services. IBM Redbooks, http://www.redbooks.ibm.com/ (accessed on 29 Dec. 2007).

Garrett, J.J., 2005. Ajax: a new approach to web applications. adaptive path Blog, 18 Feb 2005, http://www.adaptivepath.com/ideas/essays/archives/000385.php (accessed on 29 Dec. 2007).

Hinchcliffe, D., 2006. Is IBM making enterprise mashups respectable? ZDNet, http://blogs.zdnet.com/Hinchcliffe/?p=49&tag=nl.e622 (accessed on 29 Apr 2008).

Hirst, T., 2008. Geo-mashups serving the community. *GEOconnexion International Magazine*, Apr 2008, pp. 46-48.

Huang, A.W. & Chuang, T.R., 2006. Culture atlas infrastructure: a participatory and collaborative perspective, The PNC 2006 Annual Conference and Joint Meetings Aug 15 -18, 2006, the Seoul National University Library, Seoul, Korea.

Li, S., 2008. Chapter 25: "Web mapping/GIS services and applications", In 2008 ISPRS Congress Book: Advances in Photogrammetry Remote Sensing and Spatial Information Science, Chen, J., Li, Z. & Baltsavias, M. (eds.), in press.

Li, S., Guo, X., Ma, X. & Chang, Z., 2007. Towards GIS-enabled virtual public meeting space for public participation", Photogrammetric Engineering & Remote Sensing 73(6): 641-650.

MacManus, R., 2006. Mashups: who's really in control? ZDNet, http://blogs.zdnet.com/web2explorer/?p=128 (accessed on 26 Apr 2008).

Merrill, D., 2006. Mashups: the new breed of web app: an introduction to mashups. IBM DeveloperWorks, http://www.ibm.com/developerworks/xml/library/x-mashups.html (accessed on December 29, 2007).

Nivala, A.M., Brewster, S. & Sarjakoski, L.T., 2007. Usability problems of web map sites. Proc. of the International Cartographic Conference, Moscow, August 2007.

O'Brien, D. & Fitzgerald, B., 2006. Mashups, remixes and copyright law. Internet Law Bulletin 9(2), pp. 17-19.

O'Reilly, T., 2005. An article on O'Reilly Media, Inc. Website "What Is Web 2.0: design patterns and business models for the next generation of software, http://www.oreilly.com (accessed 20 Apr 2008).

Pietroniro, E. & Fichter, D., 2007. Map mashups and the rise of amateur cartographers and mapmakers. *ACMLA Bulletin*, 2007(127), pp. 26–30.

Pilgrim, M., 2002. What Is RSS, O'Reilly xml.com, http://www.xml.com/pub/a/2002/12/18/dive-into-xml.html (accessed on 20 Apr 2008).

Proto, 2007. Mashups: understanding mashup building platforms for business applications, an article by Proto Software, 140 W. 30th Street, 3E New York, NY 10001.

Ramsey, P., 2006. Mashing up the enterprise. Geospatial Solutions, Apr 1, 2006. http://www.geospatial-solutions.com/geospatialsolutions/author/authorDetail.jsp?id=34062 (accessed on 10 Mar 2008).

Reed, C. (ed.), 2006. An introduction to GeoRSS: a standards based approach for geo-enabling RSS feeds. OGC White Paper, Open Geospatial Consortium Inc. July 19, 2006.

Smith, R., 2007. SOA - enterprise mashup services, http://java.sys-con.com/read/325192.htm (accessed on 20 Apr 2008).

Schutzberg, A., 2006. Fun with GeoRSS. Directions Magazine. June 09, 2006, http://www.directionsmag.com/article.php?article_id=2197&trv=1 (accessed on 30 Apr 2008).

SYS-CON, 2008. SOA - Enterprise Mashup Services, an article published on SYS-CON Media Inc. website, http://soa.sys-con.com/read/325192_p.htm (accessed on 25 April 2008).

Taranto, T., 2007. Legalities of a mashup: a view of the collision between business and interoperability, Spatial Sciences Institute Biennial International Conference 2007, May 14-18, 2007, Hobart, Tasmania, Australia.

Van Wyngaarden, R. & Waters, N., 2007. An unfinished revolution: gaining perspective on the future of GIS. *GeoWorld*, 2007(9), pp. 22–25.

Walsh, J., The beginning and end of neogeography. GEOconnexion International Magazine, April 2008, pp. 28-30.

Wikipedia, 2008a. Wikipedia page on "Mashup (web application hybrid)", http://en.wikipedia.org/wiki/Mashup_%28web_application_hybrid%29 (accessed 25 Apr 2008).

Wikipedia, 2008b. Wikipedia page on "Web 2.0", http://en.wikipedia.org/wiki/Web_2.0 (accessed 25 Apr 2008).

Wikipedia, 2008c. Wikipedia page on "SOAP", http://en.wikipedia.org/wiki/SOAP (accessed 25 Apr 2008).

Wikipedia, 2008d. Wikipedia page on “Creative Commons”, http://en.wikipedia.org/wiki/Creative_Commons (accessed on 12 Apr 2008).