International Journal of Civil Engineering

Environmental Engineering

Technical Note

Monitoring real-time environmental information using Web 2.0 and GIServices technology

M.E. Poorazizi[1], A.A. Alesheikh[2,*]

Received: September 2009, Revised: November 2010, Accepted: February 2011

Abstract

Air pollution is a serious challenge in densely populated cities. It poses a significant threat to human health, property and the environment throughout the developed and developing parts of the world. Real-time air quality monitoring and public access to related information are the key components of a successful environmental management.

Mashups can be customized to adequately address the monitoring of such geospatially oriented challenges. Mashups combine Web 2.0 and Web services technology to facilitate data access and visualization on the Web. The use of mashups provides a cheap and effective platform for displaying real-time data.

The main objective of this paper is to assess, design and develop an environmental monitoring application based on a mashup architecture. The application uses Google Maps API (Application Programming Interface), Web GIServices (Geospatial Information Services), and AJAX (Asynchronous JavaScript and XML (eXtensible Markup Language)) to disseminate real-time air quality information through the Web. Such information can improve the decisions of the pertinent environmental organizations as well as urban settlers. The application also utilized Web client technologies such as HTML (HyperText Markup Language), CSS (Cascading Style Sheet), and JavaScript for handling the response events and providing flexibility and more interactivity. The developed mashup includes geospatial maps and features, standard business charts, node and link displays, as well as custom visual displays. All visualization components run in any Web browsers and provide a user friendly environment.

Keywords: GIS, Data interoperability, Mashups, Google maps, Web service, Geospatial data interchange format

1. Introduction

The ever-increasing population growth in urban areas amplifies certain problems, which directly or indirectly affects human daily life. Air pollution is one of the most important urban problems caused by heavy traffic, and emissions of factories that threatens human health, vegetation, and material quality. Tehran, the capital of Iran, is not an exception. To prevent or minimize the damage caused by atmospheric pollution, suitable monitoring systems are urgently needed. Such systems should rapidly and reliably detect and quantify polluting sources [1].

The urgency of affordable in-situ sensors and progressive Web 2.0 technologies are providing the fundamental infrastructure for building interactive visualization and analyzing environment for better decision making and urban planning. Such new technological infrastructures have revolutionized traditional GIS (Geospatial Information System)-based monitoring systems and provided unprecedented facilities to access, process, and visualize geospatial data [2]. Indeed, rapid developments in GeoICT (Geospatial Information and Communication Technology) has moved GIS from a static, closed, often single application environment to one that reaps the benefits of the networked environment, in particular its global and real-time accessibility [4].

Although geospatially enabled Web 2.0 initiative has introduced a major paradigm shift in how geospatial information is created, discovered, and accessed on the Internet, it cannot be considered as a true GIS because of its limited functionalities in terms of geospatial data processing and modeling [3]. These limitations can be removed through Web GIServices that facilitate discovery, accessing, processing, and visualizing geospatial data on the Web [5].

This paper elaborates on the design and implementation of a Web-based environmental monitoring application which

* *Corresponding Author: alesheikh@kntu.ac.ir*
1 Department of Geospatial Information System (GIS), K.N. Toosi University of Technology, Tehran, Iran
2 Department of Geospatial Information System (GIS), K.N. Toosi University of Technology, Tehran, Iran

facilitate disseminating air quality sensor measurements. It explores how such applications can be built on distributed GIServices and Web 2.0, relying on mashup architecture. It also describes and investigates three top geospatial data formats for mashups, namely KML (Keyhole Markup Language), GeoRSS (Geographically Encoded Objects for RSS (Really Simple Syndication)), and GeoJSON (Geographically Encoded Objects for JSON (JavaScript Object Notation)), which are mainly used to publish and visualize information on the Internet in various application domains.

2. Background

2.1. Geospatial Web 2.0

Web 2.0 provides new visualization capabilities with a new method for disseminating maps and geospatial information [6]-[7]. The use and reintegration of the technologies through open standards is the core organizing framework of Web 2.0.

From technological viewpoint, the Web is considered as a programmable platform [8]. Nowadays, most Web 2.0 services offer programmatic access by lightweight APIs. These publicly-available APIs allow programmers to easily combine services and resources from remote sources into so-called mashups that meet specific user needs. The term mashup implies easy, fast integration, frequently made possible by access to open APIs and data sources to produce value-added results [9]. Indeed, mashups are data aggregation applications, combining resources from different applications and services to create new applications and services [10]. Web 2.0 mapping platforms such as Google Maps are important example of the so-called Web 2.0 development approach that provide integrating geospatial distributed resources and visualizing them across a map.

From the map perspective, the emergence of Web 2.0 technologies has revolutionized traditional cartographic domains of visual thinking and visual communication, and opened up huge possibilities for the world of GIS, especially for geovisualization. Geovisualization can be applied to all the stages of problem-solving in geospatial analysis, from development of initial hypotheses, through knowledge discovery, analysis, presentation and evaluation [11]. The innovation and rapid diffusion of Web 2.0 products such as Google Maps has fuelled new ways of deploying geovisualization across Web platforms through a standard, easy to navigate graphic user interface [12].

2.2. Web GIServices

Web services are self-contained, self-describing, modular applications that can be published, located and invoked across the Web and perform functions that can be anything from simple requests to complicated business processes [13]-[3]. Web services provide interoperability among different resources through standard descriptions of service interfaces (WSDL (Web Service Description Language)), a standard message exchange format (SOAP (Simple Object Access Protocol)) and standard way of defining the exchange of data (XML and XSD (XML Schema Definition)) [14].

In GIS community, Web services technology is considered as enabling infrastructure for developing Web GIServices [3]. In this context, Open Geospatial Consortium (OGC) has defined a comprehensive framework of standards-compliant Web GIServices, known as the OGC Web Services (OWS) framework. The OWS framework consists of interface implementation specifications and encodings that are freely available for developers to implement [5]. It would significantly facilitate sharing geospatial data as well as access to processing services from multiple resources in and out of GIS community.

In this paper, within the OWS framework, Sensor Observation Service (SOS) has been used to disseminate air quality information in an interoperable manner. SOS is a Web service interface for requesting, filtering and retrieving sensor system information and observations [15]. SOS has three mandatory core operations including GetCapabilities, DescribeSensor, and GetObservation which used to help consumer discover and retrieve sensor and observations. Clients can get the XML-based service metadata document through the GetCapabilities operation. The DescribeSensor operation allows the clients to request information about a sensor. The GetObservation operation allows the clients to request observation data generated by a sensor. It supports a multitude of parameters and filters, which give the clients the ability to query over the sensor, time, location, phenomena, features, and measurement values of the observations. The response from the GetObservation operation is encoded in Observation & Measurement (O&M) [16].

2.3. Geospatial Data Formats

The World Wide Web is a global tool to publish, organize, and share geospatial data which play a key role in mashup development. Some new geospatial data formats have been emerged in response to the need of enabling a broad spectrum of users and developers to mash up information in a geospatial context. In the following sections, these new types of data are briefly described.

2.3.1. KML

KML is an XML-based language for managing three-dimensional geospatial data in the mapping Web applications like Google Maps [17]. It has become ubiquitous in the spatial Web as a powerful way to document and index information with a geographic reference, and displays it on maps or globe interfaces [18]. KML is an OGC standard and complementary to most of the key existing OGC standards such as GML (Geography Markup Language) [19]. A KML file specifies a set of features for display in Web 2.0 mapping platforms. These features are placemarks, geometric features, raster images, three-dimensional models, textual description, etc. Geovisualization using KML not only includes the presentation of the geospatial data on the globe, but also the control of the user's navigation in the sense of where to go and where to look.

2.3.2. GeoRSS

RSS is a Web-feed format used to publish frequently updated content such as blog entries, news headlines, or podcasts [10]. RSS becomes more and more prevalent as a way to publish and share information; it becomes increasingly important that location is described in an interoperable manner and the applications can request, aggregate, share and map geographically tagged feeds.

Geographically Encoded Objects for RSS which called GeoRSS technology is a way to encode location in RSS feeds. GeoRSS aggregates RSS and geospatial information and is supported by Web 2.0 mapping platforms. It provides flexible documents structure and standardizes the geospatial data. It is considered as an effective approach for expressing increasingly complicated geospatial data as well as non-geospatial data [20].

2.3.3. GeoJSON

JSON is a lightweight data-interchange format that is based on a subset of the JavaScript programming language [21]. It is a text-based, human-readable format for representing objects and other data structures and is mainly used to transmit such structured data over the Internet. JSON can represent data in a structured manner just as XML can, but it is far less complex. Based on JSON, GeoJSON is a new data format for encoding spatial information and non-spatial properties of geospatial features. It can be easily and quickly parsed in JavaScript, and it provides a lightweight data format that can be easily transferred. A GeoJSON object may represent a geometry, a feature, or a collection of features [22].

2.4 Air Quality Index (AQI)

Numerous environmental standards are defined in the world; in this paper AQI standard has been used. The AQI is a rating scale for reporting the ambient air pollution recorded at monitoring sites on a particular time scale (e.g., daily). The two main objectives of AQI are: (a) to inform and warn the public about the risk of exposure to daily pollution levels and (b) to enforce required regulatory measures for immediate local impact [23]. The higher the AQI value, the greater the level of air pollution and health risk.

Although the AQI itself is simply a number that reflects some aspects of air quality, in practice it is associated with color schemes, graphics, air quality category labels (e.g., "Good", "Moderate", or "Hazardous"), and various messages so that it's meaning is easily understood by the public [24]. The values of the AQI determine the air quality according to Table 1.

The AQI is estimated according to the pollution measurements in Table 2 and the following linear interpolation equation:[25]-[26]

$$I_P = \frac{I_{Hi} - I_{Lo}}{BP_{Hi} - BP_{Lo}}(C_P - BP_{Lo}) + I_{LO} \quad (1)$$

Where I_P is the index value for pollutant P, C_P is the truncated concentration of pollutant P, BP_{Hi} is the breakpoint that is greater than or equal to C_P, BP_{Lo} is the breakpoint that is less than or equal to C_P, I_{Hi} is the AQI value corresponding to BP_{Hi}, and I_{Lo} is the AQI value corresponding to BP_{Lo}" to " Where I_P is the index value for pollutant P, C_P is the truncated concentration of pollutant P, BP_{Hi} is the breakpoint that is greater than or equal to C_P, BP_{Lo} is the breakpoint that is less than or equal to C_P, I_{Hi} is the AQI value corresponding to BP_{Hi}, and I_{Lo} is the AQI value corresponding to BP_{Lo}.

Ultimately, the concentration of each pollutant is converted into AQI for the pollutant using the above formula. The pollutant with the highest AQI on a given day becomes the AQI reading for that day. It can be used to describe the impact of pollutants on human health and the environment.

The pollutant with the highest AQI number becomes the overall AQI for a particular location. The higher the AQI value is, the greater the level of air pollution becomes, and the larger the danger to human health is [24].

Table 1. AQI scales of assessment for air quality [25]

AQI Values	Levels of health concern	Colors
0-50	Good	Green
51-100	Moderate	Yellow
101-150	Unhealthy for sensitive groups	Orange
151-200	Unhealthy	Red
201-300	Very Unhealthy	Purple
301-500	Hazardous	Maroon

Table 2. AQI Categories [25]-[26]

Levels of health concern	AQI	O_3 (ppm) 8-h	O_3 (ppm) 1-h[a]	$PM_{2,5}$ (g/m^3)	PM_{10} (g/m^3)	CO (ppm)	SO_2 (ppm)	NO_2 (ppm)
Good	0-50	0.000–0.064	–	0.0–15.4	0–54	0.0–4.4	0.000–0.034	([b])
Moderate	51-100	0.065–0.084	–	15.5–40.4	55–154	4.5–9.4	0.035–0.144	([b])
Unhealthy for sensitive groups	101-150	0.085–0.104	0.125–0.164	40.5–65.4	155–254	9.5–12.4	0.145–0.224	([b])
Unhealthy	151-200	0.105–0.124	0.165–0.204	65.5–150.4	255–354	12.5–15.4	0.225–0.304	([b])
Very unhealthy	201-300	0.125–0.374	0.205–0.404	150.5–250.4	355–424	15.5–30.4	0.305–0.604	0.65–1.24
Hazardous	301-500	([c])	0.405–0.604	250.5–500.4	425–604	30.5–50.4	0.605–1.004	1.25–2.04

[a] The AQI for ozone is based on the 8-h average ozone concentration, which is computed by averaging the measured hourly ozone concentrations over an 8-h period. However, in some cases regions where the AQI is based on the hourly ozone concentrations may be more helpful. In these instances apart from the calculation of the 8-h ozone exposure based index the recorded hourly based index would provide the maximum value between them.

[b] Determination of AQI according to NO_2 limits is only allowed for values greater than 200 due to the absence of national standards.

[c] When the 8-h O_3 concentrations exceed the value of 0.374 ppm, then the AQI should be calculated by the hourly concentration.

3. Study Area

Tehran is the capital of Iran with a total area of about 700 Km^2. Mountains surround North and East part of the city. From the United Nations Report [27], urban air pollution in Tehran is among the world's worst and is increasing due to the rapid increase in population and development of industry. Air pollution and its health consequences has been a major concern for both citizens and urban planners and decision makers of this city [28]. Regarding the special geographical situation of Tehran and expansion of spatial dispersion of the pollutants which belong to stationary and mobile sources, government and citizens in this metropolitan have to use the latest technology to develop air quality management. In Tehran, in order to observe and measure air pollutants using sensor network, in-situ sensors are deployed to polluted areas. Air quality in-situ sensors measure the amount of Ozone (O3), Carbon dioxide (CO), Sulfur dioxide (SO2), Nitrogen dioxide (NO2) and Particulate Matter (PM) in polluted metropolitans [29]. Figure 1 shows the locations of air pollution monitoring stations in Tehran.

4. System Architecture and Implementation

The presented architecture is composed of AJAX enabled Web 2.0 interface as Application layer, Sensor Observation Service and Google Maps server as Service layer, and SQL Server database as Data layer (Fig. 2).

SOS is designed and implemented using .NET Web service platform. It is used to retrieve the observation data from database, perform analysis, and publish air quality information. The SQL Server 2008 database maintains spatial and aspatial information for the observed ambient air pollutant data. The Google Maps server is used as a map platform to facilitate access to air quality information and produce geovisualization on Google Maps.

4.1. JavaScript Client

The JavaScript client is the AJAX engine. It contains a user interface using dynamic HTML and CSS, a function for preparing XMLHttpRequest objects, and callback functions for handling the response events.

On one side of the user interface is the traditional Google Maps, with basic controls to pan, zoom, and switch views. This part of the interface is built when the page is loaded and the starting map location is a default location. The Google Maps API is imported as a single JavaScript file, and the desired map is added to the page using a few simple calls. The API automatically handles AJAX communication with the Google server.

Invoking Web GIService is the other part of the client functionality. When user requests air quality information the engine builds an AJAX request and sends it to the SOS, asking for geospatial file formats that contain the spatial and aspatial information of the air pollution monitoring stations. The associated callback function for this request parses the response and creates markers on the Google Maps.

Google Maps associates with KML and GeoRSS but GeoJSON is not supported yet. So for displaying the JSON formatted data, it is necessary to parse it with JavaScript functionalities and extract the spatial and aspatial information. Although KML and GeoRSS have been supported by Google Maps and it can overlay these kinds of data directly, but it provides basic presentation. Thus for more details and additional information like charts, the generated XML-based files are parsed with JavaScript DOM (Document Object Model) APIs.

4.2. Integrating Google Maps and Web GIService

As mentioned in section 2.2, we implemented a Sensor Observation Service as a Web GIService to publish air quality information. SOS serves sensor observations encoded in O&M. In order to provide observation data in KML, GeoRSS, and GeoJSON, standard SOS capabilities have been extended.

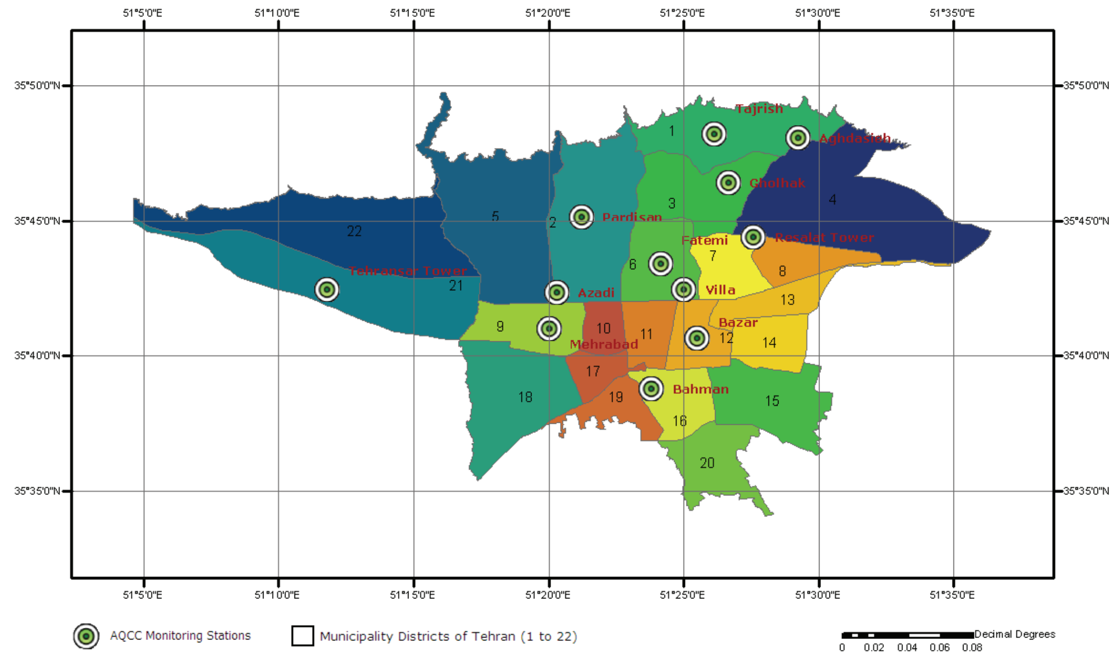

Fig. 1. The locations of air pollution monitoring stations in Tehran

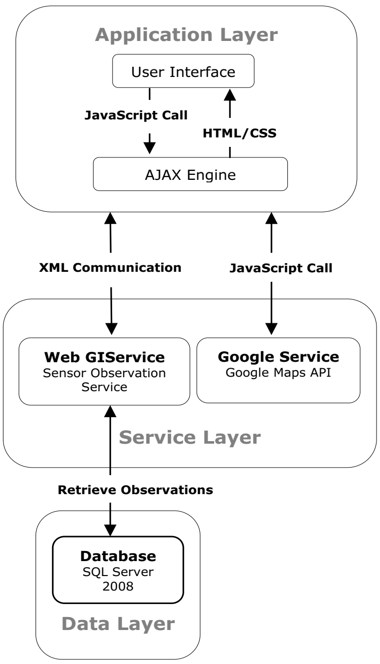

Fig. 2. Tehran Air Quality Reporter Architecture

In this context, some new features added to the SOS GetObservation request that has shown in Table 3.

As shown below, query results including sensor information, AQI and level of health conern, and geographical coordinates can be formatted in O&M, KML, GeoRSS, and GeoJSON, depending on users' preferences. The following example shows the structure of O&M encoeded data:

```
<om:Observation mlns:sos="http://www.opengis.net/sos/1.0" ...>
  <gml:name>urn:EMS:Observation:Tajrish</gml:name>
  <om:samplingTime>...</om:samplingTime>
  <om:result>
    <swe:DataArray gml:id="AirQualityInformation">
        <swe:elementCount>...</swe:elementCount>
        <swe:elementTypename="Components">...
        </swe:elementType>
        <swe:encoding>...</swe:encoding>
        <swe:values>2009-4-4T0:0:0,51.435242 ,35.806788,72
        ,Moderate</swe:values>
    </swe:DataArray>
  </om:result>
</om:Observation>
```

Air quality information can also be modeled in KML, GeoRSS, and GeoJSON encodings as shown in Table 4.

The following example gives an impression of the content and the structure of KML data file:

```
<Placemark>
   <name>Tajrish</name>
   <description>72, Moderate</description>
     <Point>
        <coordinates>51.435242,35.806788</coordinates>
     </Point>
   </Placemark>
```

The following example is an excerpt from a GeoRSS dataset containing air quality information:

```
<entry>
  <title>Tajrish</title>
  <summary>72, Moderate</summary>
  <georss:where>
    <gml:Point>
      <gml:pos>35.806788 51.435242</gml:pos>
    </gml:Point>
  </georss:where>
</entry>
```

The following example shows the content and the structure of GeoJSON data file:

```
"features": [
 { "type": "Feature",
    "geometry": {"type": "Point", "coordinates":
      [51.435242,35.806788]},
   "properties": {"prop0": "Tajrish", "prop1": "72, Moderate"}
    }
```

In response to the GetObservation request, the file format is returned in a SOAP envelope as a response to the request. After getting a response, for more details and flexibility, the client side application extracts geometry elements and attributes. The most important and commonly used geometry element is Point. Although Google Maps API supports Points geometry, but elements can also be converted to Google geometry. Having extracted and obtained geometry elements,

Table 3. Extended SOS GetObservation requests

Data format	*GetObservation* request
O&M	…responseFormat=text/xml;subtype=om/1.0.0
KML	…responseFormat=text/xml;subtype=kml
GeoRSS	…responseFormat=text/xml;subtype=georss
GeoJSON	…responseFormat=text/json;subtype=geojson

Table 4. Container elements used to encode information in KML, GeoRSS, and GeoJSON

Description	KML	GeoRSS	GeoJSON
Wrapper element of each station elements	<Placemark>	<entry>	"features"
Station name	<name>	<title>	"prop0"
AQI and level of health concern	<description>	<summary>	"prop1"
Wrapper element(s) of geographic coordinate container element	<Point>	<georss:where> <gml:Point> <gml:pos>	"geometry"
Geographic coordinate	<coordinates>	<gml:pos>	"coordinates"

these elements are plotted over the Google Maps by using "GLatLng" object and the "mapOverlay" function of the Google Maps API. By setting the returned non-geometry elements and using the "GMarker" object of the Google Maps API, this architecture also provides the attribute information such as air quality related information and charts. All these tasks are achieved by invoking the SOS using XMLHttpRequest API and JavaScript functionalities. Figure 3 shows the interaction between Google Maps server and the SOS within presented mashup architecture.

4.3. A Geospatial Mashup

The developed mashup interface is designed to be as intuitive as possible. Figure 4 illustrates the similarity between the main map interface and the standard Google Maps interface. Additional items have been added to this interface in order to add interactivity. The Google Maps standard Map Control provides zooming and panning functionalities (Fig.4 A) and its Type Control provides the different views: map, satellite image, or a hybrid of both (Fig.4 C).

A further feature added to the interface (Fig.4 C) is the ability to change the visibility of the Tehran Municipality Map from 0% (no background data visible) to 100% (background data only visible). The color of the map alters slightly as the transparency changes. SOS controls (Fig.4 B) and the legend (Fig.4 E) of the business charts are other components of the mashup application interface.

Extended SOS also generates three kinds of charts using FusionCharts Free DOM API library (http://www.fusioncharts.com) (Fig. 5). A line chart represents the air quality history of the current monitoring station; the bar chart represents the air quality index of each pollutant substance; and the hybrid chart represents the comparative exhibition and pollutant trend of arbitrary couple of monitoring stations. Line chart and bar chart are embedded in the pop-up information window of the Google Maps "GMarker" and the hybrid chart is displayed on top of the map.

AQI values displayed in charts are categorized into six numerical ranges and symbolized by different colors. Each category corresponds to a different level of health concern.

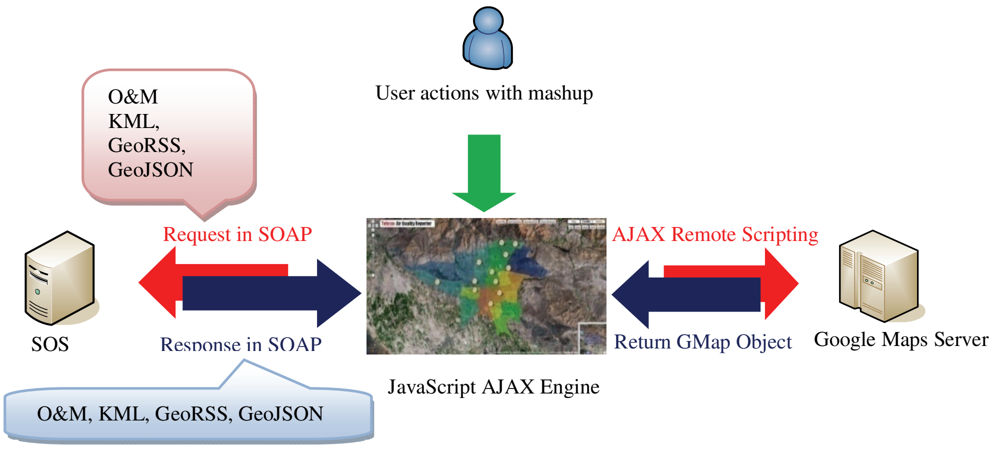

Fig. 3. Integrating Google Maps and Web GIService using AJAX

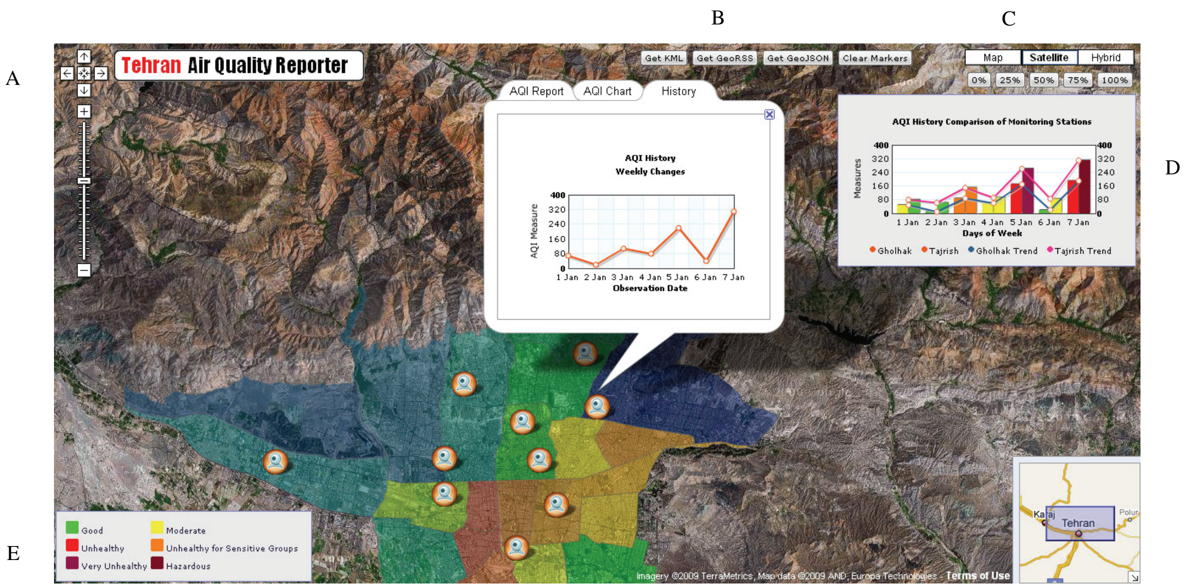

Fig. 4. The developed mashup interface includes the standard Google Maps panning and zooming controls (A), consuming SOS controls (B), the standard Map Type Control and additional map transparency controls (C), Business charts (D), and the Legend (E)

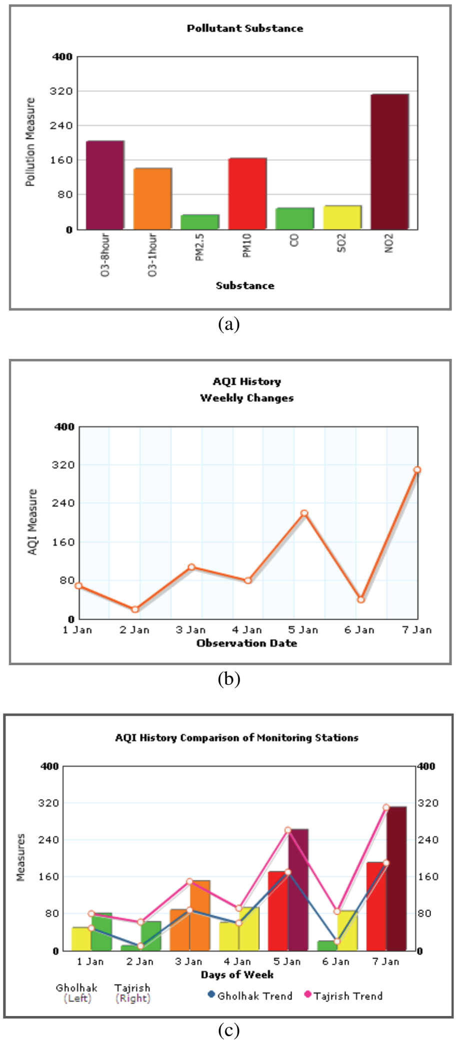

Fig. 5. Three kinds of charts displaying on mashup: (a) Bar chart, (b) Line chart, and (c) Hybrid

5. Discussion and Conclusions

Air pollution is a serious problem in thickly populated and industrialized areas in Iran, especially in Tehran. Due to increasing population, urban and industrial development, the city is faced with an air pollution problem and appropriate real-time monitoring systems are crucially needed. As the Web begins to present such information in real-time, the decision process will proceed much more rapidly and fluidly. The explosion of open APIs in the Web 2.0 world has caused a revolution in the way software is developed. Mashups are the precursor to the way software will be developed in the future. Software as a service has long been on the horizon, and Web services technology, in its many technological forms, is the enabler of cross-platform, cross-service integration that is the basis of mashups.

We have designed and implemented an environmental monitoring application based on a mashup architecture. Compared with other integrated applications, this system has several advantages. Firstly, the system is independent of platform and does not require the installation of any expensive GIS software. The users only need a Web browser to access the entire functionality. Secondly, it uses Google Maps, hence, it can use all the valuable information inherent in it, and it can significantly decrease the development cost and time. Finally, unlike other integrated systems, this system only needs very limited GIS knowledge and the user interface is based on friendly Google Maps. This advantage significantly eliminates the obstacles for the public to access the developed system.

This paper also assessed the three popular data formats that are used to visualize and publish geospatial data on the Web. How the data interchange formats evolve, and which one will be used more widespread in Web 2.0 era is a serious challenge in spatial Web domain. XML-based formats already dominated the GIS community. The description and exchange of geospatial data using KML and GeoRSS is currently a common practice. As XML is the best method for document-interchange, JSON could be the best way for data-interchange.

With the growing popularity of Web 2.0, a new data interchange format called JSON is emerging as a useful way to represent data in the business logic running on browsers. One considerable advantage of using a JSON is its ability to provide cross-domain requests while bypassing the restrictive same domain policy of the XMLHttpRequest object. On the client-side, JSON comes with a native language-compliant data structure, with which it performs much better than corresponding DOM calls required for XML processing. Finally, transforming JSON structures to presentational data can be easily achieved.

Our future research will focus on providing Web-based notification services through OGC Sensor Alert Service (SAS) and Web Notification Service (WNS) for publishing alerts from sensors, and registering users and sending notification messages to them.

References

[1] Pummakarnchana, O., Tripathi, and N., Dutta, J.: 2005, Air pollution monitoring and GIS modeling: a new use of nanotechnology based solid state gas sensors, Science and Technology of Advanced Materials 6, 251-255.

[2] Vahidnia, M. H., Alesheikh, A. A., Alimohammadi, A., Hosseinali, F.: 2009, Landslide Hazard Zonation Using Quantitative Methods in GIS, International Journal of Civil Engineering, 7 (3), 176-189.

[3] Poorazizi, E., Alesheikh, A. A., and Amini, M.: 2010, The Use of Web Services Technology for Mapping and Sharing Environmental Information in Virtual Globe Applications, International Review on Computers and Software 5 (3).

[4] Boyd, D.S., and Foody, G.M.: 2010, An overview of recent remote sensing and GIS based research in ecological informatics, Ecological Informatics (In Press).

[5] Amirian, P., and Alesheikh, A. A., and Bassiri, A.: 2010, Standards-based, Interoperable Services for Accessing Urban Services Data for The City of Tehran. Computers, Environment and Urban Systems 34 (4), 309-321.

[6] Barsky, E., and Purdon, M.: 2006, Introducing Web 2.0: Social networking and social bookmarking for health librarians, Journal of the Canadian Health Libraries Associations 27 (3), 65-67.

[7] Greaves, M., and Mika, P.: 2008, Semantic Web and Web 2.0, Web Semantics: Science, Services and Agents on the World Wide Web 6 (1), 1-3.

[8] Bugs, G., Granell, C., Fonts, O., Huerta, J., and Painho, M.: 2010, an assessment of Public Participation GIS and Web 2.0

technologies in urban planning practice in Canela, Brazil ,Cities 27 (3), 172-181.
[9] de Vrieze, P., Xu, L., Bouguettaya, A., Yang, J., and Chen, J.: 2010, Building enterprise mashups, Future Generation Computer Systems (In Press).
[10] Griffin, E.: 2008, Foundations of Popfly: Rapid Mashup Development, APress, Berkeley.
[11] Buckley, A. R., Gahegan, M., and Clarke, K.: 2004, Geographic visualization. In: R. B. Mcmaster and E. L. Usery (eds.), A research agenda for geographic information science, pp. 313-334, CRC Press, New York.
[12] Gibin, M., Singleton, A., Milton, R., Mateos, P., and Longley, P.: 2008, An Exploratory Cartographic Visualisation of London through the Google Maps API, Applied Spatial Analysis and Policy 1 (2), 85-97.
[13] Nance, K.L., and Hay, B.: 2005, Automatic transformations between geoscience standards using XML, Computer and Geosciences 31 (9), 1165-1174.
[14] Yue, P., Di, L., Yang, W., Yu, G., & Zhao, P.: 2007, Semantics-based Automatic Composition of Geospatial Web Service Chains, Computers & Geosciences 33 (5), 649-665.
[15] Na, A., and Priest, M.: 2007, Sensor Observation Service, OpenGIS® Implementation Standard, Open Geospatial Consortium Inc. <http://portal.opengeospatial.org/files/?artifact_id=26667> (visited on April 2010)
[16] Henson, C. A., Pschorr, J. K., Sheth, A. P., and Thirunarayan, K.: 2009, SemSOS: Semantic Sensor Observation Service, In Proceedings of the International Symposium on Collaborative Technologies and Systems (CTS 2009), Baltimore, Maryland, USA, 18-22 May 2009.
[17] Stefanakis, E., and Patroumpas, K.: 2008: Google Earth and XML: Advanced Visualization and Publishing of Geographic Information. In: W. Cartwright, G. Gartner, L. Meng, & M. P. Peterson (eds.), Lecture Notes in Geoinformation and Cartography: International Perspectives on Maps and the Internet, pp. 143-152, Springer, New York.
[18] Craglia, M., Goodchild, M. F., Annoni, A., Camara, G., Gould, M., Kuhn, W., Mark, D., Masser, I., Maguire, D., Liang, S., and Parsons, E.: 2008, Next-Generation Digital Earth, International Journal of Spatial Data Infrastructures Research 3, 146-167.
[19] Wilson, T.: 2008, OGC KML, OGC Document Number: 07-147r2, Ver. 2.2.0, Open Geospatial Consortium Inc., https://portal.opengeospatial.org/files/?artifact_id=27810 (visited on October 2010).
[20] Mulvenon, W.S., and Wang, K.: 2006, Integration of GIS and Educational Achievement Data for Education Policy Analysis and Decision-Making. In Proceedings of the 2006 International Conference on Digital Government Research 151, 354-355, San Diego, California, United States May 21-24, 2006.
[21] Lawrence, R.: 2004, The space efficiency of XML, Information and Software Technology 46 (11), 753-759.
[22] Butler, H., Daly, M., Doyle, A., Gillies, S., Schaub, T., and Schmidt, C.: 2008, The GeoJSON Format Specification, http://geojson.org/geojson-spec.html (visited on October 2010).
[23] Stieb, D.M., Doiron, M.S., Blagden, P., and Burnett, R.T.: 2005, Estimating the public health burden attributable to air pollution: an illustration using the development of an alternative air quality index, Journal of Toxicology and Environmental Health 68 (13), 1275-1288.
[24] Gurjar, B.R., Butler, T.M., Lawrence, M.G., and Lelieveld, J.: 2008, Evaluation of emissions and air quality in megacities, Atmospheric Environment 42 (7), 1593-1606.
[25] EPA: 1999, Air Quality Index Reporting; Final Rule, 40 CFR Part 58, American Environmental Protection Agency.
[26] Triantafylloua, A.G., Evagelopoulosa, V., and Zoras, S.: 2006, Design of a web-based information system for ambient environmental data, Journal of Environmental Management 80 (3), 230-236.
[27] United Nations: 2003, Cities and Sustainable Development: Lessons and Experiences from Asia and the Pacific, Economic and Social Commission for Asia and the Pacific, United Nations, New York.
[28] Sohrabinia, M., and Khorshiddoust, M.: 2007, Application of satellite data and GIS in studying air pollutants in Tehran, Habitat International 31 (2), 268-275.
[29] Alesheikh, A. A., Oskouei, A.K., Atabi, F., and Helali, H.: 2005, Providing Interoperability for Air Quality In-Situ Sensors Observations Using GML Technology, International Journal of Environmental Science and Technology 2 (2), 133 -140.