2018 International Conference on Advanced Science and Engineering (ICOASE), Kurdistan Region, Iraq

Mapping Environmental Sounds Using Google Map (Acoustic Maps)

Hanan A. Taher
Department of Information Technology and Management
Polytechnic University
Duhok, Iraq
hanan.taher@dpu.edu.krd

***Abstract* - Modern lives are dominated by what people see, and so there are many travel guides described wonderful places to visit. However, the sounds of these places are part of the experience. In some cases, what people hear is more important and interesting than what they see. This paper presents a new technique that gives an approach to combine map locations with their environmental sounds. It identifies places with unique sounds and encourages people to become interested in what they hear. The system is a collaborative data input, which allows public contribution in recording and archiving. Users are allowed to participate and upload their own sounds onto an efficient mapping system (Google map). The idea is to develop an audio map application; it is a great way to encourage people to gather audio and plot sounds to locations. The system is equivalent to Google Street View but with embedding audio will be Audio Street View.**

Keywords- Mapping, Google Map, Acoustic maps.

I. INTRODUCTION

E-reader and tablets are becoming more popular as physical media on the other hand is declining. This is the impact of internet technologies. Digital Maps have become more common, and physical maps have been replaced essentially by digital maps. This is because digital map offers the most up-to-date contents. City environments sound and maps can be associated with each other; users may want to know precisely what the environment sound look like when selecting locations in map. When people are looking for some place to live, they might want to know whether it is a noisy or a quite area. Likewise, system beneficiaries might be City Planner, Environment and Urban Planning Ministry, and Housing and Urban institutes. They might use this data to have an idea where is the noisy area in order to do some noise controller. Moreover, it's a useful material and good to be submitted and published online for researches.

The purpose is to build a web-based tool for exploring acoustic map data. Users will be presented with a map and will be able to select location to hear what the sound like. This is done by sorting the data on a server to make it accessible over the internet, and then build extra tools for uploading the user's own data. This encourages people to record their own data and upload it into the same server and over the time we can collect more audio data.

II. MAPPING INTERFACES

This section deals with using an actual mapping interface. A tool such as Google Maps is needed to view cities from space. Google Maps received Street View; this innovation feature is used in this system to combine audio with Street View.

There are two main mapping services available online, one of which is the Google Maps service and the other is the Bing Maps Service run by Microsoft. Google Maps has been in service for much longer than Bing Maps and so naturally will generally have better coverage, but not necessarily more up to date coverage. There is more than just a map for these two services, as each of these offer additional features such as satellite images, hybrid views (satellite images with map layer on top), and also panoramic street level imagery [1, 2]. Both Mapping services offer very clear and up to date maps, but Google Maps offer slightly more detailed maps, with public walkways and paths included in the maps.

III. GIS SOFTWARE

A. Definition of Geographic Information System

GIS stands for Geographical Information System, simply, GIS is a business information management system that helps users to capture, analyze, and present all types of spatial or geographical information on a map[3].

The combination of spatial mapping and database (GI System) was produced because of the need to locate the attributes of the spatial data [4]. The available tools and data sets have not satisfied the requirements of people who involved in spatial information wok. People start to demand for more developed tools to help them in analyzing the geolocation information. Users currently are demanding simulation of varicose geolocation processes rather than just static analysis [4]. They are demanding three dimensional view of the surface of the world (for example) rather than a two dimensional visualization. The developers always are looking further as the system is no longer enough.

IV. WEB TECHNOLOGY FOR IMPLEMENTING DYNAMIC WEBSITE

There are various types of web technologies that are used to build web solution including client-side technologies, server side technologies, and frameworks among others [5]. This section describes the chosen technologies that have

978-1-5386-6696-8/18/$31.00 ©2018 IEEE

been used in developing this system together with the reasons of choosing them.

A. Client And Server Model

According to Bill Fote [6], the "client" refers to web browsers like Google Chrome, Internet Explorer, Firefox, etc. in the other hand, the "server" is a web application server that processed the web request and send pages to the client at the remote location. The architecture of Client-Server model used in this system is shown in Fig. 1.1. The communication between client and server technologies is necessary for a smooth development of the web application and to enable applications being developed to be more useful and worked on all kinds of devices for users [7].

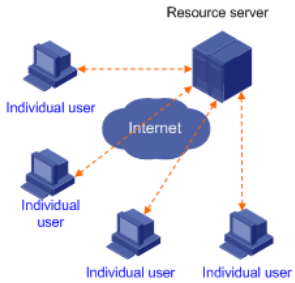


Table 1.1 The structure of Client-Server Model

B. Server Side Programming Languages

Server-side processing is used to return pages to client and process data that has been input by the user. It is also responsible of interacting with the constant storage in the system such as databases or files (Carl, 2009). The market for server-side programming solution has become a crowded one in today's fast paced world of web design. Language like Python, Ruby, PHP, Java, and many more are the most widely used server-side scripting languages. In fact, even though, they were initially emerged using different of programming languages they have their futures and functionalities that facilitate and make web application to communicates smoothly by standardizing data with JSON, XML, RSS, or files [8].

C. Choice of Programming Languages

Java is the most suitable scripting language choices for developing this software as it is faster than any other language and it is stable, dependable, and was designed to be a language that could run on any device. The chosen server platform is Apache Tomcat 7.x, which supports Java versions 6 and later. This supports JSP Spec 2.2. The language used is Java, specifically Java Servlets and Java Server Pages for server-side processing. The website is connected to a MySQL database version 5.6.x, which contains audio and GPS data. Client-side rendering take place using HTML5, CSS3 and Javascript. The diagram below shows the trends at which Java language is used by web developers [9].

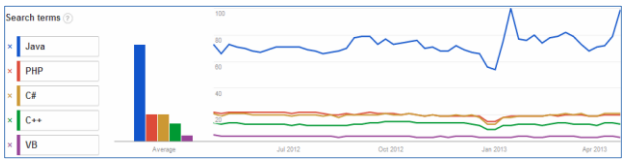


Fig. 3.1: Programming Language Popularity July 2012 – April 2013(Source: Google Trend Tools, Justin Cooney, 2013)

V. CHOSE OF DATABASE

MySQL has been popular and used database in most software systems; because of its speed and ease of use. PostgreSQL is more suitable for developers with the background from oracle database. Many developers perceives that MySQL is simple and easy to use than PostgreSQL due it complexity and restriction in terms of data storage and retrievals (Mediawiki, 2009). Fig 4.1 below shows that MySQL is the highest of open sources database distribution in 2011. For developing this system MySQL database system has been chosen.

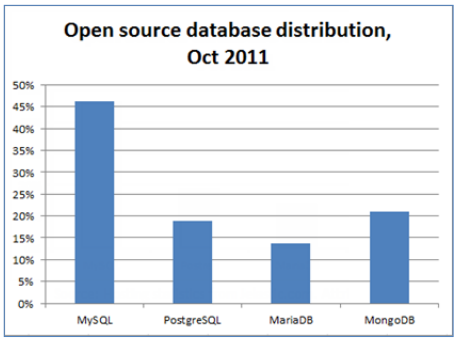


Table 4.1Open Sources Database Distribution Oct 2011

VI. SYSTEM DESIGHN

A. Model View Controller (Mvc)

Acoustic maps system built on the traditional Model-View Controller (MVC) architecture, which separated the design into three subdirectories: models, views and controllers. The three-tier architecture is one of the basic model architecture of software development.

The code for each layer can be cleanly split away from the others. For example, the presentation layer has no direct communication with the data access layer, and it can only contact with the business layer. System architecture and software tools are illustrated in Fig. 5.1. Briefly, the core advantage of using the MVC architecture is that it help to systemize and splits the code of Acoustic maps System into the separate file. This benefit of separation of concerned makes the development easy to extend, quickly change, and facilitate maintenance as it allows modification of component independently in existing functionality (Surguy, 2014).

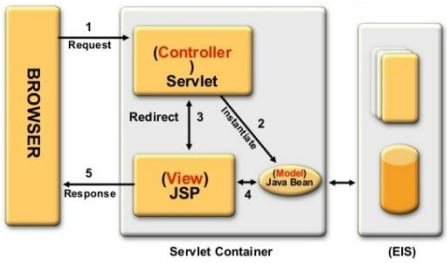
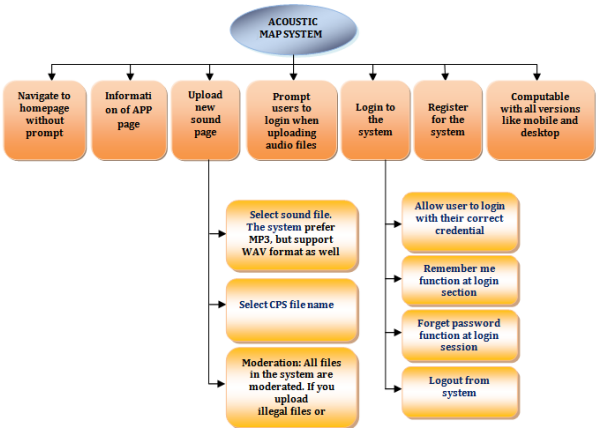


Fig. 5.1: MVC Design Pattern

As described above, the infrastructure comprises an Apache Tomcat Web server connected to a MySQL database. Server-side programming takes place in the form of JSPs/Servlets. Client-side programming is based around HTML5, CSS3 and Javascript.

B. Webside Structure



VII. SYSTEM IMPLEMENTATION

Once design stage is completed, a development stage is continued to the implementation phase. There is various definition of implementation in the development as referred by Roseland (2013), implementation can be defined as the method which involves transferring the idea from concept to the reality.

A. Servlet Implementation To Upload Files

Acoustic map aims to create a huge collaborative database of audio recordings which are companied by GPS data, and used them to locate the audio in a map. The user can contribute by uploading his own records, if recordings are geotagged, then the map will be marked with the location of recording. Ideally, a web page form is created that enables the user to upload audio and GPX files together.

After a successful uploading, the files will be saved to database. JSP and Servlet are used for uploading files to server. JSP is used to setup a form, and for showing the result of file upload operation while Servlet is used to handle file upload request. The following objects is used to specify the maximum size of an upload file and specify maximum size of a HTTP request which contains the upload file, system checks if there is upload request.

```
public class UploadServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

//private static final String DATA_DIRECTORY = "/Users/HananTa
private static final int MAX_MEMORY_SIZE = 1024 * 1024 * 2;
private static final int MAX_REQUEST_SIZE = 1024 * 1024 * 900;
```

```
// Check that we have a file upload request
boolean isMultipart = ServletFileUpload.isMultipartContent(request);

if (!isMultipart) {
    return;
}
```

If there is upload request, system creates a factory for disk-based file items, constructs the folder where uploaded file will be stored, and creates a new file upload handler as showing in the following script:

```
Create a factory for disk-based file items
kFileItemFactory factory = new DiskFileItemFactory();

Sets the size threshold beyond which files are written directly to
disk.
tory.setSizeThreshold(MAX_MEMORY_SIZE);

Sets the directory used to temporarily store files that are larger
than the configured size threshold. We use temporary directory for
java
tory.setRepository(new File(System.getProperty("java.io.tmpdir")));

constructs the folder where uploaded file will be stored
tring uploadFolder = DATA_DIRECTORY;
etServletContext().getRealPath("")+ File.separator + DATA_DIRECTORY;
Create a new file upload handler
vletFileUpload upload = new ServletFileUpload(factory);

Set overall request size constraint
oad.setSizeMax(MAX_REQUEST_SIZE);
```

All file formats (MP3, WAV, GPX) are converted to lower case so that lower/upper case character is accepted. Java class is used for parsing GPX files and thereby extracting all information needed (lat, long, ele, time, and file name) and stores them in the database. JBDC connection pool is used to open a database server connection (Fig. 1.1). Java.sql package is used as it contains JDBC classes.

```
Class.forName("com.mysql.jdbc.Driver");
String myDbUrl = getServletContext().getInitParameter("DB_URL");
String DBuser = getServletContext().getInitParameter("DB_USER");
String DBpassword = getServletContext().getInitParameter("DB_PASSWORD")
java.sql.Connection con = DriverManager.getConnection(myDbUrl, DBuser,
                                                      DBpassword);
Statement st = con.createStatement();
```

Fig. 1. Configure a JBDC connector

The data recourse is an audio recording made while taking walking tours around the city. The recording should be using special in–ear microphones that record sounds in the way that they are actually heard. The audio is

accompanied by GPS data that can be used to locate the audio on map.

For data validation, the system accepts only two audio file formats which are MP3, WAV. For the GPS data the system accepts a GPX file format. Both files should be uploaded together. If user tried to upload one file at a time, user will get a notification message saying "Please upload your Sound and GPS Files". There is no limitation in audio file size. If user clicked 'Upload files' button without choosing any files, system will display 'Please upload your sound and GPS files' message. Successful "Upload New Sound" will navigate user to another page and will get notification saying "Your files has been uploaded!" message. New road will be drawn on map to present the sonic environment of this location. Failed "Upload New Sound", will navigate user to another page and will get notification saying "Error in uploading" message.

B. Draw Polyline By Coordinates And Improve Snap-To-Road Quality.

Google Maps API offers a class for drawing roads on a Google Map called Polyline. The coordinates needed to create a ploryline are retrieved from database. Firstly all latitude/longitude points are saved into database using Java class. Then, in order to improve snap-to-road quality, one point from each 20 points is selected using 'getpoints' function (Fig. 2.1).

```
var map = new google.maps.Map(mapCanvas, mapOptions);
var flightPlanCoordinates = [<%= getpoints()%>];
var flightPath = new google.maps.Polyline({
    path: flightPlanCoordinates,
    geodesic: true,
    strokeColor: '#FF0000',
    strokeOpacity: 0.2,
    strokeWeight: 50.0
});
```

Fig. 2.1 Creating a polyline using Geolocation

In order to increase the accuracy of the road drawing on map, an average of (latitude/longitude) sample is taken for a given location. Fig. 2.2 is a screen shot of the website displays the map when using sampling, and Fig. 2.3 displays the map when points to skip are 0. As shown from the figures, drawing the road without sampling cased in drawing inaccurate road.

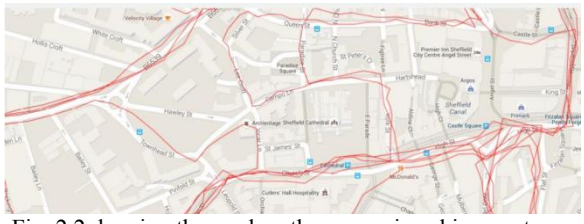


Fig. 2.2 drawing the road on the map using skip counter

Fig. 2.3 drawing the road on the map without skip counter

Using polyline properties, a 50.0-pixel-wide red polyline is created. Fig. 2.4 is pictorial evidence for the success implementation of this feature.

Fig. 2.4 User-created polylines

Latitude/longitude pairs are sorted out in ascending order by time. Thus points are drawn consecutively on the map. User can navigate to 'Home' page to display the map and see the uploaded audio file being auto-geotagged and marked as a red path (Fig. 2.4)

C. Extract Specific Segment of Audio File

When user clicks on any waypoint of polyline, the system should urn audio record of this location. Thus, a specific segment of audio record needs to be extracted. It is not easy to extract data from hours of audio recordings.

This feature implemented as following: Each audio file has **'creation time'** and is saved in the database. The audio file is accompanied by GPX file. Each waypoint in the GPX file has 'latitude', 'longitude', and '**time**' attributes which are saved in the database. By subtracting '**creation time**' from '**time**' and dividing over 1000 (the result must be in milliseconds), segment start offset is determined. Segment duration is 2 minutes and stored in XML file as 'Play_seg'. These two minutes of audio sound where decided to be 1 minute before and one after the clicked location. There is a situation where a mouse clinked could be at the beginning of the audio file when there is no recording for the first 1 minute. The offset will be negative which is unacceptable. Math.max method is used to solve this problem, if the result offset is negative, 0.0 is used as segment start offset (see Fig. 3.1).

```
long t1 = rs2.getTimestamp("creationtime").getTime();
long t2 = rs.getTimestamp("time").getTime();
long offset = (t2 - t1) / 1000;
Startoffset = Math.max(0.0, offset - (playSeg/2));
//out.println("playSeg = "+playSeg);
String DATA_DIRECTORY = getServletContext().getInitParameter("dataPath
String data = File.separatorChar + DATA_DIRECTORY + File.separatorChar
AudioSrc = sfn.substring(sfn.indexOf(data)+1)+"#t="+Startoffset+","
                                        +(Startoffset+(playSeg/2
```

Fig. 3.1 Setting start offset and end point of audio

D. Setting A Click Event Listener To Open Info Window

From Google Maps API, addListener( ) method is used. The method takes an event to listen for (user click), and a function to call when the specified event occurs (open info window). Likewise, infow window object and SetContent( ) method used from Google Maps API to create and open info window at different location upon map event which is user clicks. SetContent( ) method used to set the info window content. AJAX request used to get the data (lat, lon, start offset); the return will be set in the info window content.

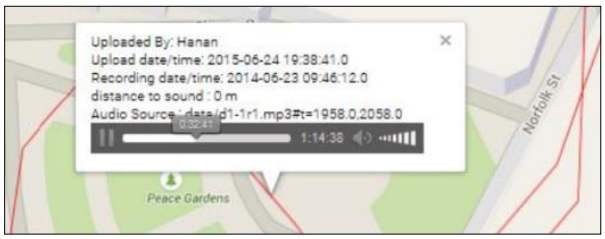


Fig. 4.1: audio segment runs in info window content

When polyline is clicked by users, an info window will open and system will run the audio recording sound of this location for 2 minutes. It will display: the audio file name, GPS file name, the user name who uploaded these files, the file creation time, and distance to the road. Fig. 4.1 is approving of successful implementation of this feature.

The sound waves usually travel outward from its point of origin through the air in narrow and wide circular tubes. Thus, users may click around the location and still hear the same sound. The system allows users to click around highlighted road with distance up to 50 meter to hear its environmental sound. The following function in figure 4.2 shows a function implemented to calculate distance from a clicked location to nearest sound.

```
public static String getQuery(String slat, String slng, int maxdist) {
    String query
            = "SELECT filename,lat,lng,ele,time,round(6372797.560856 * acos(cos( radians(" + slat + ")) * cos(radians(lat))"
            + "* cos( radians(lng) - radians(" + slng + ")) + sin(radians(" + slat + "))"
            + "* sin( radians(lat)))) AS distance"
            + " FROM GPXinfo HAVING distance < " + maxdist + " ORDER BY distance;";
    return query;
}
```

Fig. 4.2: Query statement finds nearest sound location

Latitude/longitude points retrieved from database. 'maxdist' is the max distance to sound (50 meters) is saved and obtained from XML file as in the following instruction.

```
String maxdist = getServletContext().getInitParameter("distance");
```

The same query sorts out the results in ascending order using ORDERD BY Clouse (by default, the ORDER BY Clouse sorts data in ascending order). Results less than (50 meter) are selected and are sorted out in ascending order.

Commonly altered values are saved in MXL file; to improve the program readability and assist maintenance. Thus, the software is able to be customizing without editing the program source code. XML file is created to store database configuration, max distance to sound, session timeout, location to store uploaded files, latitude/longitude to skip, and the segment duration ('Play_seg'). These input data values can be changed without the need to change the source code and run the application again. All what needs is to open the XML file and change to desired data (Fig. 4.3).

description	The maximum distance in meter t
param-name	distance
param-value	50
description	Points to skip when drawing
param-name	PointsToSkip
param-value	20
description	The length of the sound segmen
param-name	play_segment
param-value	120

Fig. 4.3: XML file nods and its contents

VIII. SYSTEM TESTING

The system has been tested on a real-world example area using 20 hours of audio recording (audio and GPX files). An end-to-end testing was done to ensure the system works and to ensure that it is error free, accessible, and user-friendly. Taking into consideration the testing principles, the application has been tested by people who are outside of the development process.

System users were chosen to test the product, users who need this application and concerned about it. According to the user feedback results, the system implemented successfully. The system collects data from a wide area

without spending expensive money. On the other hand, any software system may have weak points, thus, some of them are out of the software developer control. Weak point of this system is that the audio files cannot be controlled by the administrator, to avoid audio files which do not include environmental sound, before being uploading and locating to the map. However, administrator can delete the files from database after being imbedded in map. Another weak point is the difficulty in detecting poor recording quality by Java API.

REFERENCES

[1] G. Maps. (2016). *http://maps.google.co.uk/ - Last accessed 3/7/18*.

[2] B. Maps. (2018). *http://www.bing.com/maps/*

[3] O. Survey. *WHAT IS GIS: https://www.ordnancesurvey.co.uk/support/understanding-gis/what-is-gis.html*.

[4] C. M. Gold, "What is GIS and What is Not?," *Transactions in GIS,* vol. 10, pp. 505-519, 2006.

[5] S. S. Shreerang Hegde, "A SURVEY ON THE LATEST WEB TECHNOLOGIES," *International Journal of Technical Research and Applications,* vol. 3, 2015.

[6] B. Fote. (2013). *Client-Side Vs. Server-Side Code: What'S The Difference?*

[7] H. Zhang, "Architecture of Network and Client-Server model," *arXiv preprint arXiv:1307.6665,* 2013.

[8] C. L. Cervinschi and D. Butucea, "Integration of web technologies in software applications. Is Web 2.0 a solution?," *Database Systems Journal,* vol. 1, pp. 39-44, 2010.

[9] J. Cooney. (2013). *Reviewing the Popularity of 5 Commonly Used Server-Side Programming Languages*.