DYNAMIC PROJECTION OF DATA ON MAPS BASED ON TIMELINES:

CLIENT SIDE

A Thesis
Presented to the
Faculty of
San Diego State University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Computer Science

by
Swagath Manda
Summer 2012
SAN DIEGO STATE UNIVERSITY

The Undersigned Faculty Committee Approves the

Thesis of Swagath Manda:

Dynamic Projection of Data on Maps Based on Timelines: Client Side

Carl Eckberg, Chair
Department of Computer Science

Joseph Lewis
Department of Computer Science

Carmelo Interlando
Department of Mathematics

May 4, 2017
Approval Date
Copyright © 2012
by
Swagath Manda
All Rights Reserved
DEDICATION

I would like to dedicate my thesis to Prof. Carl Eckberg for all his support and motivation.
ABSTRACT OF THE THESIS

Dynamic Projection of Data on Maps Based on Timelines: Client Side
by
Swagath Manda
Master of Science in Computer Science
San Diego State University, 2012

One of the main uses of GIS is to help explain or understand and analyze events and trends on a particular region or set of geo-spatial data. However in many cases the data is time sensitive and the data in the system needs to change dynamically in accordance with time and developing such systems requires considerable effort. Google Maps has in many ways revolutionized and simplified GIS and made it available to the masses, however powerful the service may be, it lacks a simple method to enable users to create and publish geospatial data on their maps and more so when it comes to creating dynamic time sensitive data. Motivation for this thesis is to develop an intuitive GIS web framework that enables the users to create and publish GIS using the free and readily available solution provided by Google Maps.

This website is developed keeping in mind teachers and students, and common users in such a way that it can be used by both experts and naïve users to create powerful Geographical Information Systems with minimal effort and time. The framework enables the creation of TimeLine Map, which has a time scale ruler and a Google map integrated. The data is projected on the map dynamically based on the timescale. The users can slide through the time ruler to see the corresponding data being projected on the map dynamically. The advanced users can tweak the solutions provided to further customize the Time Map, give filters, color coded place marks and enable touch events as demonstrated with the examples – British Empire and US Presidents TimeLine Map.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT ............................................................................................................................... v</td>
</tr>
<tr>
<td>LIST OF FIGURES ............................................................................................................... viii</td>
</tr>
<tr>
<td>CHAPTER</td>
</tr>
<tr>
<td>1 INTRODUCTION ..................................................................................................................... 1</td>
</tr>
<tr>
<td>1.1 Purpose of the Project ...................................................................................................... 2</td>
</tr>
<tr>
<td>1.2 Scope of the Project ......................................................................................................... 2</td>
</tr>
<tr>
<td>2 LITERATURE SURVEY ............................................................................................................ 3</td>
</tr>
<tr>
<td>2.1 Review of Existing Systems ........................................................................................... 3</td>
</tr>
<tr>
<td>2.1.1 Animated Time Lapse Videos ....................................................................................... 3</td>
</tr>
<tr>
<td>2.1.2 Google Earth – Time Slider ....................................................................................... 4</td>
</tr>
<tr>
<td>2.1.3 Dipity ........................................................................................................................... 6</td>
</tr>
<tr>
<td>2.1.4 ArcGIS.com – Time-Enabled Maps ............................................................................. 7</td>
</tr>
<tr>
<td>2.2 Limitations of Existing Systems ..................................................................................... 7</td>
</tr>
<tr>
<td>3 APPLICATION ...................................................................................................................... 8</td>
</tr>
<tr>
<td>3.1 Implementation ................................................................................................................. 8</td>
</tr>
<tr>
<td>3.2 Technology Used ............................................................................................................. 9</td>
</tr>
<tr>
<td>3.2.1 Timeline and Time Map Libraries ............................................................................. 9</td>
</tr>
<tr>
<td>3.2.1.1 Creating Timelines .............................................................................................. 10</td>
</tr>
<tr>
<td>3.2.1.2 Timelines Parameters ......................................................................................... 11</td>
</tr>
<tr>
<td>3.2.1.3 Core Classes ........................................................................................................ 11</td>
</tr>
<tr>
<td>3.2.2 Google Maps JavaScript API ..................................................................................... 11</td>
</tr>
<tr>
<td>3.2.3 jQuery ....................................................................................................................... 12</td>
</tr>
<tr>
<td>3.2.4 HTML5, jQuery Mobile, PHP ................................................................................... 13</td>
</tr>
<tr>
<td>4 CODE SNIPPETS ................................................................................................................ 14</td>
</tr>
<tr>
<td>4.1 Link to the API ................................................................................................................. 14</td>
</tr>
<tr>
<td>4.2 Create a Timeline Element .......................................................................................... 14</td>
</tr>
<tr>
<td>4.3 Call Timeline ............................................................................................................... 14</td>
</tr>
</tbody>
</table>
4.4 Synchronization of Band in Timelines ............................................................15
4.5 Add Events.......................................................................................................16
4.6 Basic Event Attributes .....................................................................................16
4.7 Placemark.........................................................................................................17

5 SCREENSHOTS..........................................................................................................20
5.1 jQuery Mobile UI.............................................................................................20
5.2 Google Map Controls.......................................................................................22
5.3 Google Spreadsheets........................................................................................23
5.4 TimeLine Maps................................................................................................25

6 LIMITATIONS AND FUTURE ENHANCEMENTS ...............................................34

7 CONCLUSION AND OBSTACLES ..........................................................................36

BIBLIOGRAPHY ....................................................................................................................38
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Time lapse video of nuclear tests since 1945.</td>
<td>4</td>
</tr>
<tr>
<td>2.2</td>
<td>Mongol Empire time on Google Earth Gallery.</td>
<td>5</td>
</tr>
<tr>
<td>2.3</td>
<td>Time Slider on Google Earth.</td>
<td>6</td>
</tr>
<tr>
<td>2.4</td>
<td>Timeline of NASA space shuttle launches.</td>
<td>6</td>
</tr>
<tr>
<td>2.5</td>
<td>NASA space shuttle launches – shuttle launch sites projected on map.</td>
<td>7</td>
</tr>
<tr>
<td>3.1</td>
<td>Application architecture.</td>
<td>8</td>
</tr>
<tr>
<td>3.2</td>
<td>Application components.</td>
<td>8</td>
</tr>
<tr>
<td>3.3</td>
<td>Application components – technologies and API used.</td>
<td>9</td>
</tr>
<tr>
<td>3.4</td>
<td>Timeline – bands.</td>
<td>10</td>
</tr>
<tr>
<td>3.5</td>
<td>Timeline – parameters.</td>
<td>11</td>
</tr>
<tr>
<td>3.6</td>
<td>jQuery plugin to embed Google Map on webpage.</td>
<td>12</td>
</tr>
<tr>
<td>4.1</td>
<td>Place marks example.</td>
<td>18</td>
</tr>
<tr>
<td>5.1</td>
<td>Index page of the timeline map website on computer.</td>
<td>20</td>
</tr>
<tr>
<td>5.2</td>
<td>Website optimized on mobile devices – iPhone, iPad.</td>
<td>21</td>
</tr>
<tr>
<td>5.3</td>
<td>UI options optimized on mobile devices with jQuery Mobile.</td>
<td>21</td>
</tr>
<tr>
<td>5.4</td>
<td>Zoom.</td>
<td>22</td>
</tr>
<tr>
<td>5.5</td>
<td>Pan.</td>
<td>22</td>
</tr>
<tr>
<td>5.6</td>
<td>MapType.</td>
<td>23</td>
</tr>
<tr>
<td>5.7</td>
<td>Generating Google spreadsheet ID.</td>
<td>23</td>
</tr>
<tr>
<td>5.8</td>
<td>Publish spreadsheet and obtain ID key.</td>
<td>24</td>
</tr>
<tr>
<td>5.9</td>
<td>Create timeline maps from Google spreadsheets.</td>
<td>24</td>
</tr>
<tr>
<td>5.10</td>
<td>Timeline created from spreadsheet.</td>
<td>25</td>
</tr>
<tr>
<td>5.11</td>
<td>Time scale.</td>
<td>25</td>
</tr>
<tr>
<td>5.12</td>
<td>Time scale – day, week, year, and decade.</td>
<td>26</td>
</tr>
<tr>
<td>5.13</td>
<td>Create timeline from KML files.</td>
<td>27</td>
</tr>
<tr>
<td>5.14</td>
<td>Upload KML and generate timeline map.</td>
<td>27</td>
</tr>
<tr>
<td>5.15</td>
<td>USA presidents – birthplaces, tenure, famous quotes.</td>
<td>28</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

A geographic information system (GIS) is an information system that is designed to work with data referenced by spatial or geographic coordinates. It is both a database system with specific capabilities for spatially-reference data, as well as a set of operations for working with data [4].

One of the main uses of GIS is to help explain or understand and analyze events and trends on a particular region or set of geo-spatial data, performing geospatial analysis.

There are several approaches and methods for creating GIS Systems. They can be broadly classified into Application based systems such as ESRI ArcView, Oracle Spatial, PostGIS etc and web-mapping systems, such as the solutions provided by Google Maps, Bing Maps, Open Layers. The applications based GIS tools, which were first developed by ESRI since 1969 are used in most military, government institutions. They have a high degree of customization, several geospatial analysis features and capability to display data on a wide range of projection models, enable the creation of customized shape files that can be projected as layers on maps, which make them very powerful tools that are indispensable in the professional sectors. However they do have limited usage among the general public due to the complexity in using them, licensing woes and the relative difficulty in developing GIS systems using them.

The web-mapping solutions are a relatively new approach for GIS, developed by Keyhole Inc., which was later acquired by Google to produce the GIS products :-Google Maps and Google Earth. Web-based GIS is free of cost, and is simple to use, readily available, and updated constantly. Thereby they have widespread usage among the public and have become prominent tools for geospatial analysis in the non-commercial and public domain. A testament to this is the phenomenal success of the Google Maps and Google Earth applications that have revolutionized the way the public perceives and uses GIS. These systems enable users to view Maps without the need to download or update any shape files, or install any particular software and thereby provide simple and elegant solutions. They are
highly dependent on the availability of a network connection and lack many of the features present in application-based solutions that limit their use in professional sectors.

1.1 PURPOSE OF THE PROJECT

The data to be projected on maps, in most cases, have just spatial constraints, such as geographic coordinates or are shape files. Both Web and Application based GIS systems can display these types of data on maps in a suitable manner. In cases where the data is time sensitive, i.e. the data has time constraints along with spatial constraints, the system needs to change data on maps dynamically in accordance with time, is such a way that the time-sensitivity of the data is not lost. Currently such systems can be developed using Google Earth Time Slider in Flight mode [10] and the Time Span feature in ESRI Arc View [1]. However these systems require a considerable amount of effort to develop. The existing app based, web based GIS do not provide a simple method to create and view such maps.

Motivation for this thesis is to develop an intuitive GIS web framework that enables users to create and publish their own GIS that is capable of displaying time sensitive data dynamically on maps; with considerable ease. These types of maps will be referred to as timeline maps henceforth.

1.2 SCOPE OF THE PROJECT

This website is developed keeping in mind teachers and students, and common users in such a way that it can be used by both experts and naïve users to create powerful timeline maps. I have to used Google Maps JavaScript API [7] and Timeline JavaScript Library [17] to create the timeline maps. The users can slide through the timeline ruler to see the corresponding data being projected and removed from the map dynamically, based on the existence of the data on the timeline.

The website enables users to easily develop timeline maps, using Google Spreadsheets or by uploading KML files. With the former method, users can create and publish timeline maps quickly with minimal effort, however they do not have many options to customize the maps created; with the latter, they have a wide range of options to customize the maps as demonstrated with the British Empire and US Presidents timeline examples as part of the thesis.
CHAPTER 2

LITERATURE SURVEY

We have chosen Google Maps, a web-mapping solution as a base for developing these GIS systems due to its ubiquitous accessibility and wide spread presence in the public sector. The availability of the Google Maps JavaScript API allows further customizations of the maps and adds features as demonstrated in the US Presidents example.

Google Maps has in many ways revolutionized and simplified GIS and made it available to the masses. However powerful the service may be, it lacks a simple method to enable users to create and publish Geo-Spatial data on their maps and more so when it comes to creating dynamic time sensitive data. Motivation for this thesis is to develop an intuitive GIS web framework that enables the users to create and publish GIS using the free and readily available solution provided by Google Maps.

2.1 REVIEW OF EXISTING SYSTEMS

Timelines provide a visual medium of analyzing the flow of events and help in understanding and explaining events better. They can be created to represent several events in history, or track the progress of events in real time, biographical events of a person’s life, track science, weather patterns and so on.

In some cases the nature of the data to be presented is also in spatial in nature, such as weather, earthquakes. In these cases a better representation of the event can be achieved by projecting the data on the maps based on their time of occurrence using time scales or timelines. There are several existing systems that can be used to create such timeline maps. The advantages and limitations of these systems are discussed below.

2.1.1 Animated Time Lapse Videos

Time Lapse Videos can be used to visually represent the flow events in a chronological order. When this data is mashed on a map along this sequence we will essentially be able to create a timeline map video. This method, though it can be used to achieve the desired objective, requires a lot of effort. No framework as such is used to
generate such timeline maps. They are suitable only for scenarios with a limited amount of data available, and lack flexibility in analysis, generation and projection of data. The user’s will not be able to interact with these systems either.

The example in Figure 2.1 [19] shows a time lapse video in YouTube depicting nuclear detonations since the beginning of the atomic age [19]. Flashes on the map indicate nuclear tests and progresses through time are indicated through the year counter in the top left corner of the video.

![Figure 2.1. Time lapse video of nuclear tests since 1945. Source: YouTube, Nuclear testing 1945-1988 complete video HD. YouTube, http://www.youtube.com/watch?v=WAnqRQg-W0k, accessed June 2011, n.d.](image)

This method, though effective, has no reusability and is not scalable.

### 2.1.2 Google Earth – Time Slider

Google Earth [9] is a virtual globe, mapping the Earth with satellite and aerial images. Detailed KML files with geographic coordinates and time stamps are required to create timeline maps using Google Earth. Either a Google Earth browser plugin or the standalone
application Google Earth is required to view the time line map animations. TimePrimitive function is able to parse through the time stamps and generate a time slider, which controls the data on the map. This is an effective and comprehensive solution for providing timeline maps; however the complexity of the KML file required to generate such a map makes it less user friendly and time consuming. This application provides portability and scalability by allowing the users to store the data in KML files, and parses through these files to extract and project data [15]. See Figure 2.2 [15] for the Mongol Empire on Google Earth Calendar, and Figure 2.3 [8] for the Time Slider on Google Earth.

2.1.3 Dipity

Dipity [1] is a website which allows users to create timelines online. It is a freely available service that allows users to create and publish content online. The data can be projected as several formats such as a timeline, on a map, or as a list of events. It provides a simple, user-friendly interface to the users for creating and sharing timelines. It allows the addition of multimedia content with timestamps and geo-tags that can be projected on a map. However it does not allow true dynamic projection of data on maps.

The screenshots found in Figures 2.4 [3] and 2.5 [2] show a Dipity Timeline representing NASA Space Shuttle Launches.
2.1.4 ArcGIS.com – Time-Enabled Maps

Time-enabled maps are ESRI’s solution to generating time line maps. Users can create and share maps and online and have access to ArcGIS Online, which is ESRI’s repository of maps, applications and tools. Time sliders are created by including an image service layer that store information about a dataset changing over time. This application is not open source and is licensed by ESRI [1].

2.2 LIMITATIONS OF EXISTING SYSTEMS

The existing systems discussed above either lack portability, flexibility or a simple interface to generate, publish and view timeline maps. In some cases as with ArcGIS web based solution, they are licensed software that cannot be used freely. With systems like Dipity, they do not offer true dynamic projection of data.

This project is aimed at developing an open source GIS tool that addresses the issues that are present in these existing systems. We use Google Maps API, Timemap and timeline javascript library to create a framework that can be used to generate, publish and share timeline maps with ease.
CHAPTER 3

APPLICATION

3.1 IMPLEMENTATION

Google Maps JavaScript API [7] and jQuery [13], PHP [16] are used for server side interaction. Timeline.js and timemap.js JavaScript is used at the Client End to generate timeline and control the data that is to be projected. HTML5 [18] and JQuery Mobile [17] are used for the user interface design and to make the application portable to mobile devices. Figure 3.1 gives a basic overview of the architecture of the project. The Client side used to generate the TimeMap and project data, whereas the Server end is used to populate the maps with data from Google. Figure 3.2 shows the components used at the client end and the server end of this tool.

Figure 3.1. Application architecture.

Figure 3.2. Application components.
Google maps JavaScript API allows embedding of Google maps in web pages. It provides utilities for manipulating and adding content to the map.

JQuery is a cross-browser JavaScript library that simplifies HTML document traversing, event handling, and Ajax interactions for rapid web development. The Graphical User Interface is designed using jQuery Mobile and HTML5. It provides a standardized set of form controls and UI widgets on webpages. We use HTML5 for file upload process along with PHP, a general-purpose scripting language that is especially suited for server-side web development.

Timemap.js and timeline.js are JavaScript libraries that can be used to create timelines and load datasets on online maps. Timeline contain one or more Bands, which can be panned. Data on the map is changed dynamically based on the timeline.

### 3.2 Technology Used

The applications components, including the technologies and API used can be found in Figure 3.3.

![Figure 3.3. Application components – technologies and API used.](image)

#### 3.2.1 Timeline and Time Map Libraries

Timemap.js is a JavaScript library that can be used with online maps: Google Maps, Bing and other similar mapping solutions. It allows users to load datasets on to a map and a timeline simultaneously.
The datasets can be in KML or JSON formats which are supported by timemap.js parsers. To handle custom data formats, the users can add customized parser modules to provide data to the library.

TimeMap is a timeline containing one or more Bands, which can be panned infinitely by dragging with the mouse pointer, using the mouse scroll-wheel or the keyboard's arrow buttons.

3.2.1.1 CREATING TIMELINES

A timeline is implemented as a div element that contains inner div elements as its bands. The band div's are cropped and positioned relative to the timeline div (see Figure 3.4).

![Figure 3.4. Timeline – bands.](image)

A band contains several inner elements that implement various parts of the band. As a band is panned, it is shifted horizontally or vertically, carrying all of its visual elements along. When either end of the band approaches the visible area, the band re-centered, its coordinate origin is changed, and then its various visual elements are re-painted relative to the new coordinate origin. All of this paging is done as seamlessly as possible so that the user experiences smooth panning.

A band is responsible for supporting panning as well as coordinating its various sub-components. The band also takes an event source which provides events to be displayed in that band. Different bands can have different event sources.

Based on the portion of the timeline in the visible region, corresponding data is projected on the map. This flexibility allows for timeline mashups, as demonstrated with the British Empire example.
3.2.1.2 Timelines Parameters

Non-Duration Event, also known as an Instant event, is focused on a specific time such as a person's date of death. The time of the instant event is marked by the icon, or more specifically, by the middle of the icon.

A Duration Event is an event that occurs over a period of time, such as the dates of a war. Duration events are shown with a solid tape, and no icon. See Figure 3.5 for an example of timeline parameters.

![Figure 3.5. Timeline – parameters.](image)

3.2.1.3 Core Classes

These are the base classes of the TimeMap library data:

- TimeMap - The TimeMap object holds references to timeline, map, and datasets.
- TimeMapDataset - The TimeMapDataset object holds an array of items and dataset-level options and settings, including visual themes.
- TimeMapItem - The TimeMapItem object holds references to one or more map place marks and an associated timeline event.

3.2.2 Google Maps JavaScript API

By using the Google Maps API, it is possible to embed Google Maps site into an external website, on to which data can be overlaid. The Google Maps API is free for educational use and thereby we can use this to create an open source tool for the generation of time maps.

To use the service, the user is required to obtain an API key by registering the domain and server details of the webpage where the service is intended to be used with Google. The APIs provides flexibility to embed maps in webpages and flash-based apps. It offers support for embedding virtual globes (using Google Earth plugin), street view panorama and location based services, access to geocoding, directions, and other information.
Some other notable features available are place-marks positioning on maps, KML overlays shape and data, tile overlays, ground overlays, traffic overlays: adding real-time traffic information, driving directions and so on. More information on these features can be found on the Google Maps documentation page.

The latest version of this API (version 3) has been used in this project, as it is faster and more applicable to mobile devices, as well as traditional desktop browser applications [5].

### 3.2.3 jQuery

jQuery is a JavaScript library. The jQuery framework allows the creation of powerful and dynamic web pages through its modular approach. The jQuery library is available online and is open source. It is stored as a single JavaScript file. The jQuery file needs to be referenced by the webpage to make use of the library. The version hosted by Google or Microsoft can be used as an external reference by the webpage to reduce data traffic on primary web page server, or a local copy can be referenced internally to improve performance and load times.

jQuery has a large user community, with a wide range of plugins, such as GMap plugin, a Google maps jQuery plugin, which can be used to embed Google maps in the webpage (see Figure 3.6 [12]).

![Google maps jQuery plugin basic example](image)

**Figure 3.6.** jQuery plugin to embed Google Map on webpage. Source: GMAP, What is gMap? gMap, [http://gmap.nurtext.de/](http://gmap.nurtext.de/), accessed March 2011, n.d.
3.2.4 HTML5, jQuery Mobile, PHP

HTML - Hyper Text Markup Language describes the content of webpages using markup tags. HTML5 supports for cross-platform mobile applications. It simplifies the development of Webpages and helps produce code, which is more portable to mobile devices.

JQuery Mobile framework helps in building uniform UI across web and mobile platforms with. The framework can be used to makes websites portable to mobile devices. It has module that leverage HTML5, CSS3, and JavaScript features.

PHP- Hypertext Preprocessor is a general-purpose scripting language used in server-side web development to create dynamic web pages. PHP is used for storage, retrieval and processing of KML files on the Rohan server in this project [15].
CHAPTER 4

CODE SNIPPETS

4.1 LINK TO THE API

The code below is used to link the timeline.js API to our webpage. A local copy can be downloaded onto the host server and be given as an internal reference to speed up load and response time.

```html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" 
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html;charset=UTF-8" />
    ...
    <script src="http://static.simile.mit.edu/timeline/api-2.3.0/timeline-api.js?bundle=true" type="text/javascript"></script>
    ...
</head>
<body>
    ...
</body>
</html>
```

4.2 CREATE A TIMELINE ELEMENT

The code below is used to create a Timeline Element in the webpage:

```html
<div id="my-timeline" style="height: 150px; border: 1px solid #aaa"></div>
<noscript>
</noscript>
```

4.3 CALL TIMELINE

The code below is used to create a Timeline on the webpage. The Band Info parameter width, height, interval unit, interval pixels are initialized. The function to create
the timeline is called when the webpage is loaded initially or when the webpage repainted or resized later.

```html
<body onload="onLoad();" onresize="onResize();">
...
</body>
var tl;
function onLoad() {
  var bandInfos = [
    Timeline.createBandInfo({
      width:          "70%",
      intervalUnit:   Timeline.DateTime.MONTH,
      intervalPixels: 100
    }),
    Timeline.createBandInfo({
      width:          "30%",
      intervalUnit:   Timeline.DateTime.YEAR,
      intervalPixels: 200
    })
  ];
  tl = Timeline.create(document.getElementById("my-timeline"),
                   bandInfos);
}

var resizeTimerID = null;
function onResize() {
  if (resizeTimerID == null) {
    resizeTimerID = window.setTimeout(function() {
      resizeTimerID = null;
      tl.layout();
    }, 500);
  }
}
```

### 4.4 SYNCHRONIZATION OF BAND IN TIMELINES

The code below is used to keep multiple bands of a timeline in sync. We can add multiple bands to a timeline to provide information of two or more different sets of timeline.
In cases where the data is related, it makes sense to synchronize these timelines. We use the `bandInfos[1].syncWith` parameter to achieve this.

```javascript
bandInfos[1].syncWith = 0;
bandInfos[1].highlight = true;

function onLoad() {
    var bandInfos = [
        Timeline.createBandInfo({
            width:          "70%",
            intervalUnit:   Timeline.DateTime.MONTH,
            intervalPixels: 100
        }),
        Timeline.createBandInfo({
            width:          "30%",
            intervalUnit:   Timeline.DateTime.YEAR,
            intervalPixels: 200
        })
    ];
    bandInfos[1].syncWith = 0;
    bandInfos[1].highlight = true;

    tl = Timeline.create(document.getElementById("my-timeline"),
    bandInfos);
}
```

### 4.5 Add Events

The code below is used to add click events to a Timeline on the webpage. They can be used to trigger dialog boxes giving additional information when user clicks on a layer.

```javascript
eventSource:    eventSource,
date:           "Jun 28 2006 00:00:00 GMT",
```

### 4.6 Basic Event Attributes

The following are the basic event attributes. They are required to create any event on a given timeline:
✓ start – start time in full date format.
✓ latestStart - for imprecise beginnings - time in full date format.
✓ earliestEnd - for imprecise ends - time in full date format.
✓ end – end time in full date format.
✓ durationEvent - XML takes: "true" or "false". Only applies to events with start/end times.

4.7 PLACEMARK

Place marks can be used to drop reference points, shapes and layers on maps using the Google Maps API [11]. The following properties for the new place mark:

✓ **Name** - for the place mark.
✓ **Description** - including HTML text.
✓ **Style, Color** - Choose a color, scale or size and opacity for the place mark icon.
✓ **View** - Choose a position for the place mark.
✓ **Altitude** - Choose the height of the place mark as it appears over terrain. Applicable for virtual globes, not maps.
✓ **Icon** - Click the icon for the place mark.

Each map marker can have a unique custom icon, share a common custom icon, or use the default icon. It has options to set the map type - street, satellite view, size of the map, zoom level etc [6].

The types of place marks are:

✓ Polygon
✓ Polyline
✓ Marker
✓ Overlay
✓ Multiple Marker

The code below is used to create the place marks on a Google map. The Polygon, Polyline, Marker, Overlay and Multiple Marker place marks are created as shown in the screenshot in Figure 4.1 [17].

```javascript
{
    start : "2010-05-01",
    placemarks: [
    {
        polyline: [
```

{lat: 48.22467264956519, lon: 16.32568359375},
{lat: 47.517200697839414, lon: 18.984375},
{lat: 45.84410779560204, lon: 15.9521484375},
{lat: 48.22467264956519, lon: 16.32568359375}
},
),
),
{
point: {
lat: 48.22467264956519,
lon: 16.32568359375
}
},
{
point: {
lat: 47.517200697839414,
lon: 18.984375
}
},
{
point: {
lat: 45.84410779560204,
lon: 15.9521484375
}
title : "Multiple Placemark Item",
options : {
  description: "I am large, I contain multitudes."—Walt Whitman<br>My info window is anchored based on the first loaded item.",
  theme: "blue"
}
CHAPTER 5

SCREENSHOTS

5.1 JQUERY MOBILE UI

The following screen shot shows the homepage of this application. The UI is created using jQuery Mobile and is optimized for viewing other mobile devices as shown in Figures 5.1 [14] and 5.2 [14].

Figure 5.1. Index page of the timeline map website on computer. Source: K. MATHIVANAN AND S. MANDA, Timeline map. San Diego State University, http://www-rohan.sdsu.edu/~mathivan/, accessed March 2011, n.d.
Using jQuery Mobile, the UI is buttons, drop down menus are designed to provide the native look and feel of the platform that is used (see Figure 5.3 [14]).
5.2 Google Map Controls

Google maps contain UI elements to control the map and allowing the user to interact with the map. The following controls are available to the user.

Zoom controls (see Figure 5.4 [14]) displays, map area viewable using a slider with ‘+’ and ‘-‘ buttons to zoom in and zoom out of the map respectively. Pan control (see Figure 5.5 [14]) allows panning the map, move the area of the map that is viewable.


The MapType control (see Figure 5.6 [14]) lets the user toggle between the different MapTypes such as:

- Map - default road map view.
- Satellite - Google Earth satellite images.
- Hybrid - mixture of normal and satellite views.

5.3 GOOGLE SPREADSHEETS

Figures 5.7 [14] and 5.8 [14] show how the user can generate the spreadsheet ID. Timeline maps can be generated using the ID, the map refers to the spatial data present in the spreadsheet.


Figure 5.9 [14] shows a spreadsheet with spatial data and time data that is used to populate the time map. The columns present in the spreadsheet can’t be changed dynamically.

- Latitude
- Longitude
- Title
- Description
- start
- end

5.4 Timeline Maps

Figure 5.10 [14] shows a TimeMap being generated using the Google spreadsheet ID.

![Timeline Map](image)


Figure 5.11 [14] shows the time scale used in TimeMap used to handle data on timescales of varying time ranges. The associate code with the time interval units is also given below.

```javascript
TimeMap.init({
    bandInfo: [
        {
            width: "85%",
            intervalUnit: Timeline.DateTime.DAY,
            intervalPixels: 210
        }
    ],
```

Figure 5.12 [14] shows the time scale being used on units of day, week, year, and decade.


Figures 5.13 [14] and 5.14 [14] show a KML file being uploaded to populate the data on the TimeMap.

Figure 5.15 [14] shows the US presidents timeline being generated using a KML file.

Figure 5.15. USA presidents – birthplaces, tenure, famous quotes.

Figure 5.16 [14] shows the US President’s timeline, and the usage of start and date parameters to create events representing the tenure of the president. It also shows the themes and tags used to categorize the party to which the President belonged – Democratic and Republican and visually represent them using red and blue colors. On clicking an event on the timeline or clicking a place mark, a pops up window gives more information. The Picture, of the president a link to his profile on Wikipedia, along with a famous quote from the president is displayed (see Figures 5.17 [14] and 5.18 [14]).

```
{  "title": "John F. Kennedy",  "start": "1953-01-20",  "end": "1961-01-20",  "point": {    "lat": 42.3316,    "lon": -71.1216  },  "options": {    "theme": "blue",    "description": "tags: ['Democrat']",    "tags": ["Democrat"]  }}
```


Figure 5.19 [14] shows the British Empire timeline, the lines are events representing the tenure of start and end of British rule on a region. The themes red and yellow are used to indicate new conquests, expansion and freedom of the colonies. On clicking an event on the timeline or clicking a polygon shape mark, a pops up window gives more information on the British colony.

The TimeMap in the British Empire timeline, used multiple band of different timescales – a small scale and a large scale timeline. These bands are synchronized and allow the user to get an overview of the overview of the flow of events using the small scale timeline, or use the large scale timeline to get an in-depth picture (see Figures 5.20 for map [14], 5.21 for the start and end parameters in TimeMap [14], and 5.22 for Polygon shapes and themes [14]).

Figure 5.23 [14] shows the US States acquisition timeline, the lines are events representing the tenure of start date of acquisition of a region. The red polygons are used to indicate new conquests and states acquired. The polygon gives the boundary of the state acquired. On clicking an event on the timeline or clicking a polygon shape mark, a pops up window gives more information on the State.

```javascript
{
    start: "1757-08-02",
    end: "1947-08-15",
    polygon: [
        {lon: 98.525391, lat: 10.401377 },
        {lon: 99.52734, lat: 11.867350 },
        {lon: 99.052734, lat: 12.83147 },
        {lon: 99.052734, lat: 13.83271979 },
        {lon: 98.895738, lat: 14.325672 },
        {lon: 98.895738, lat: 15.804541 },
        {lon: 97.52812, lat: 16.729502 },
        {lon: 97.52812, lat: 17.138470 },
        {lon: 100.107422, lat: 20.550508 },
        {lon: 100.107422, lat: 21.799903 }
    ]
}
```

The TimeMap in this example uses only one band in the timeline. It used CSS to create vertical panes for the timeline and for the map.

The KML file used to generate the US States acquisition TimeMap (see Figure 5.24 [14]), has the following data in it:

- Place mark – Polygon with multiple coordinates giving rise to s shape form.
- Timespan - parameter in KML Files with begin, end to create events on the timeline.
- Place mark IDs.
CHAPTER 6
LIMITATIONS AND FUTURE ENHANCEMENTS

The web based GIS solutions created do not have cross compatibility with ESRI products. This is a major limitation as most professionals seeking GIS systems use ESRI. Switching to web based solutions will require ground up development of all systems, and shape files rather than allowing the user to utilize the shape files that were already available from their earlier ESRI systems. An advantage the users will have with web based GIS systems is the cross compatibility with other web based systems (E.g. Bing Maps) using standard formats to represent data, like KML.

KML systems do not support the concept of separate shape files. Shapes are emulated in the form of polygons using multiple coordinates. Though this allows the user to gain more accuracy in the shapes while scaling the maps, it requires more data to represent the shapes and causes a significant overhead during development.

With the present implementation of the tool, only certain predefined parameters can be entered in the Google spreadsheets, and data can be imported only from KML or .csv files. Further enhancement in this area would be to allow users to provide custom defined data fields and allow SQL data to be imported, provide support for Google Fusion tables that let users store, share, and manage tables. Providing support for SQL and Fusion table backend will also provide the tool with improved capabilities to handle very large volumes of data and non-spatial data such as images; which it currently lacks with KML .CSV file based systems.

The Google Maps and the Timeline API also lacks certain functionalities present in ESRI, such as flashing shape files, changing units dynamically to suit the data being projected at present. This issue is evident with the Presidents example where timescale can’t be modified dynamically for varying length of presidential tenures. We have used a workaround in this tool to give the user control to change the time scale to optimize data projection on time maps. We will be able to add more functionality to the tool as the Timeline API matures and we have more features available.
Given the number of projects produced using ESRI Map Objects and its widespread usage, a similar application framework to help develop timeline maps can be developed.
CHAPTER 7

CONCLUSION AND OBSTACLES

The tool developed as part of this thesis is an intuitive GIS web framework that enables users to create and publish their own TimeLine Maps, with considerable ease. These maps can display time sensitive data dynamically on maps. The data with spatial and time constraints is projected dynamically on maps such that the time-sensitivity of the data is portrayed visually.

These maps can be used as an effective visual medium to educate students on time based events that have a geographic connection as demonstrated with the examples: British Empire, US Presidents Tenure and US States Acquisition Timeline Maps. They can be used effectively in history, geology and others areas involving spatial data. The tool offers portability by publishing the maps online using Google Maps thereby making them available to a large user base. The users do not require any technical expertise to use this tool.

Several obstacle were encountered during various phases of the project, they are discussed below.

To use the Google Maps API one has to obtain an API Key by registering with Google. This key is unique and changes for every domain and server the code is hosted on. The key needs to be updated in the back end accordingly.

Due to lack of compatibility with ESRI products, which has extensive usage in the GIS domain, all the data need to be recreated for this project. Existing shape files and other spatial data built with ESRI tools could not be used and had to be created specifically for this project. Since Google Maps systems do not support the concept of separate shape files they have to be emulated in the form of polygons using multiple coordinates. This involved time consuming effort while creating the polygons for the British Empire timeline. However the polygon files created for this project can be stored in KML files and be reused in other applicable areas, more resources will be available as the user base increases.

The Timeline API lacks ability to scale the timeline by dynamically assessing the timeline range that is being analyzes by the user at present. Due to this, when handling very
large or small time ranges the timeline UI needs to be adjusted accordingly, to provide a convenient user experience. This is achieved in this project using timescale control. In worst case scenarios which involve a mixture of varying time ranges, as in the US Presidents example, the timescale can’t be modified dynamically.

We had encountered several compatibility issues making the application work across multiple browsers. They were addressed by switching to jQuery Mobile which provided cross compatibility with multiple browsers. HTML5 and CSS3 were used for to resolve the issue of optimizing the application for multiple mobile devices.
WORKS CITED


WORKS CONSULTED


