Geographic Visualization Of Event Popularity: A Proof-Of-Concept Prototype Utilizing The Last.fm Web Services API And Google Earth

Patrick Tod Colegrove, Jessica Gallagher, and Lee Dash*

Abstract—The authors report on the design and implementation of a proof-of-concept prototype viewer of localized social music networking event data, parsing data returned from calls to the Last.fm Web Services API, geocoding and transforming the XML data into a KML file suitable for vieing using a geobrowser such as the Google Earth viewer. A total of ten events from the geographic area surrounding the coordinates of Philadelphia within a 2 km radius were visualized, with event popularily, multiple events collocated at a single venue, as well as upcoming/past events clearly represented in an intuitive and accessible format for the end-user.

Index Terms—Information Visualization, Three-dimensional displays, Markup Languages, Visualization techniques and methodologies

1 Introduction

Geographical visualization has advanced greatly from where it first began. For centuries, people have created and used maps to represent a place relative to other locations. The first images may have been as simple as drawing lines to indicate a river or shading to represent mountains. As cartographers began to become better through the use of more precise instruments (e.g. a compass), the maps became much more intricate. Further than that, as technology has advanced, more sophisticated geographical images can be rendered.

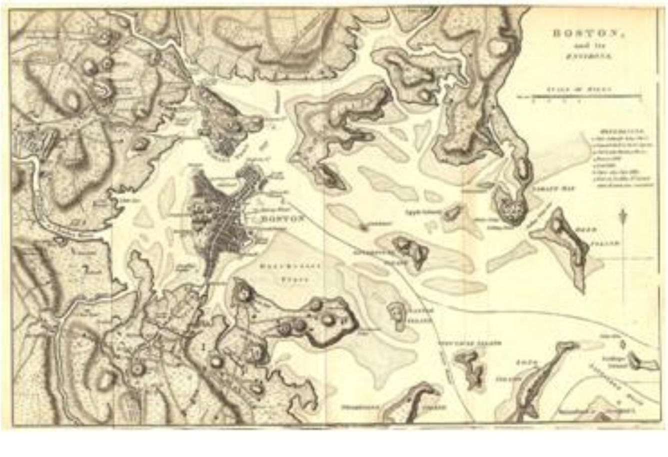


Fig. 1. An early cartographic representation of Boston, MA, dated to the 18th century [2].

1.1 About Last.fm

For the purposes of this project, we decided to geographically visualize the popularity of local music events using data from the social music networking site Last.fm [4]. Last.fm is a robust online community made up of music enthusiasts from across a wide spectrum of genres. The site allows users to report their listening habits through a process called 'scrobbling' which enables Last.fm to make recommendations about new music to listeners, as well as to connect users with other people of similar musical taste. In this paper, the authors report details of a proof-of-concept prototype that leverages the existence of the Last.fm application programming interface (API) [5], developing a Google Earth visualization that bridges the gap between the online community of Last.fm listeners and the geographic world of the physical location of event venues. The following describes Last.fm, its Web Services API and implementation specifics of its use in the visualization.

A popular social music networking website, Last.fm brings together artists and music listeners from across hundreds of music styles and genres. The core functionality of the site is its ability to log what artists and songs individual users are listening to. This is achieved through a process known as 'scrobbling' in which users run a small application on their computer which detects the song currently being played in a standard media player (i.e. iTunes, Windows Media Player, etc.) and reports that information back to Last.fm. Over time the site builds a detailed profile about the user based on their listening preference, which is then used to make strong recommendations to users about other bands or types of music they might enjoy. It can also introduce users to others who have similar musical tastes, or rate the level of musical compatibility between two given users. The site employs a number of common social networking features including the ability to customize user profiles with an avatar (photo) and additional information, as well as the ability to add friends and send messages to others.

In addition to tracking artists and music listeners, the Last.fm site also tracks scheduled events and artists that are playing (concerts, shows, etc.), along with specific local venue information. The visualization described was conceived as adding a link between the event feature of the Last.fm site and the physical world of the scheduled event venues. An event on Last.fm can be made up of multiple artists but takes place at a specific geographical venue on a specific day and time. This information about the event is posted on the event page along with links to the band's website and other particulars, such as where tickets may be purchased online. The event system not only makes users aware of upcoming shows, but also lets them declare to their friends and the Last.fm community their plans to attend the event. When users mark on their profile that they plan to attend an event, their name and avatar appear on the corresponding event page stating that this particular user plans to attend. This information is of particular interest to the online community, and is frequently used by the local community engaged in the process of selecting which of multiple events to attend within a limited span of time available to them; the simultaneous display of that information was by the authors as filling a real need in the current offering of Last.fm, with its representation fundamental to this visualization. Users can also post 'shouts' on individual event pages asking questions or expressing excitement about the upcoming event.

1.2 Google Earth

Google Earth enables the rapid visualization and overlay of abstracted representations of data and geographic imagery that share a common geographic coding; in Figure 2 the physical flight path between the origin and destination cities has been represented with arcs connecting the cities – the arc itself an abstraction of each flight’s takoff from one city, ascent to a higher altitude reaching its apex midway to the destination city, followed by its descent and arrival. Stacked bars and various colors have additionally been used to signal different aspects of the data set.

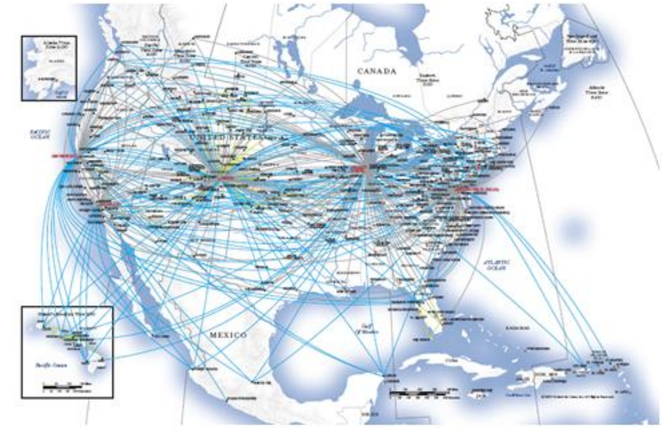


Fig. 2. A geographic representation of the flight paths and interconnections of United Airlines flights visualized in Google Earth. The blue arcs represent a two-dimensional rendering of each flight’s three-dimensional path through the intervening space between origination and destination airports [3].

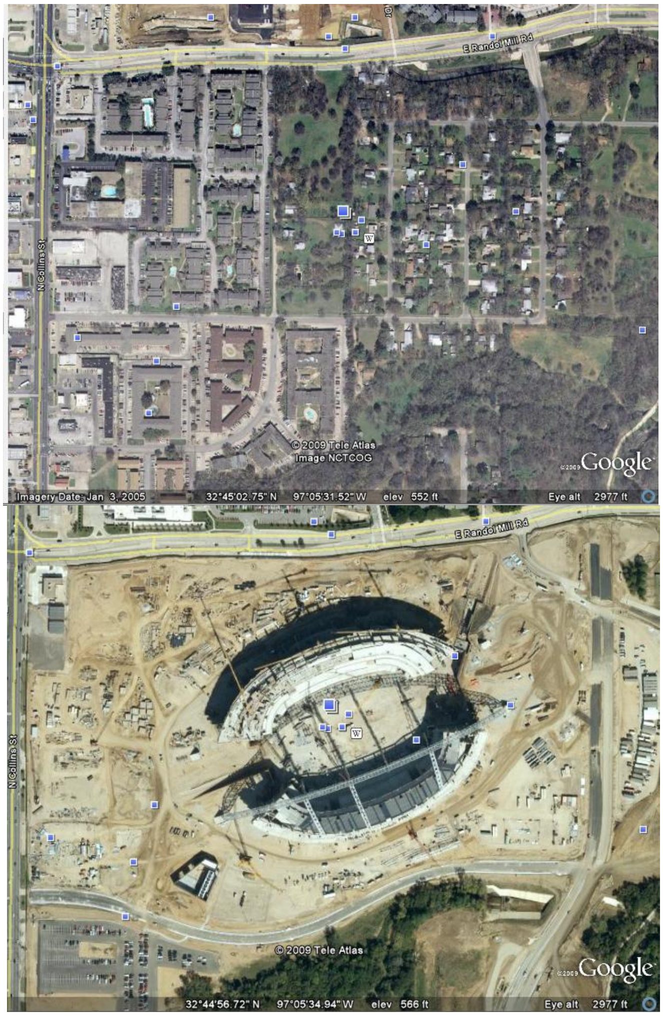
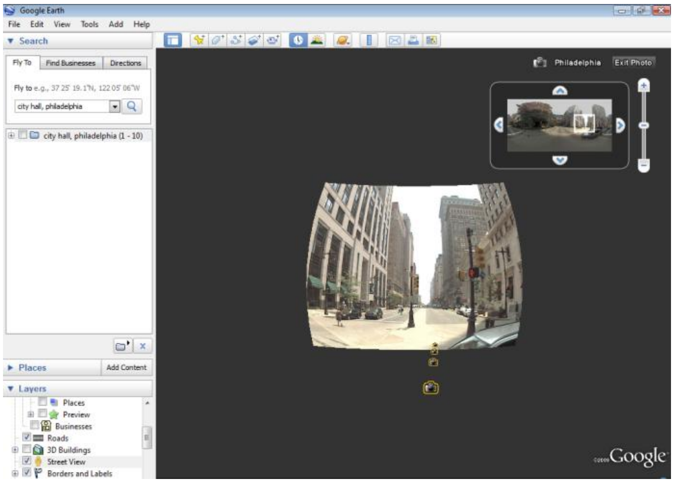


Fig. 3. Google Earth imagery that captures before and during construction timeline of the new Dallas Cowboys stadium in Arlington, Virginia; top image dated April 1, 2005, bottom October 26, 2007.

The geobrowser, such as Google Earth, represents a huge advancement in geographical representation; Google Earth builds an entirely three dimensional replica of the earth by downloading satellite data. This data has been stored in a remote server and is updated with new images when needed. Since the first version of Google Earth was available, it has adapted and evolved to the needs of the user. It is possible now for users to interact with personal data to create their own rendition of a particular place – a possibility that is leveraged in this project.

Google Earth has several advantages to its two dimensional counterparts. First is the ability to fully control the picture. The earth is displayed on several different axes, which allows the user to rotate the globe whichever way is desired. The zoom feature allows the user to drill down to a specific place (such as one’s own home) and see details about that location (like the car in the driveway). Resolution does limit clarity when the user zooms into the maximum, but “this zoom facility allows small-scale



Fly to e.g., 37 25' 19.1"N, 122 05' 06"W

city hall, philadelphia

city hall, philadelphia (1 - 10)

Fig. 4. Street view layer in Google Earth of Center City, Philadelphia.

topographic and geological features to be instantly placed in a broader regional context" (Lisle 2006).

Another advantage of the program is the ability to incorporate arcs connecting separate geographical locations, enabling the user to simulate a 'flight' from one location to the next. The program also incorporates places of interest already into the features and can be selected or deselected. In addition, data has been stored from previous satellite data downloads and with the push of a button the user can travel back to previous times of a specific spot. For example, the new Dallas Cowboy stadium that has been created in Arlington, Texas was recently completed. The previous view of it in 2005 shows a wooded area with dozens of buildings.

Another significant feature of Google Earth is the program's ability to bring in other features from other Google programs. Pictures of an area can be clicked on from the visual field. This will generate an information display box, which can include links to featured websites, additional resources, or even a link to have Google Maps retrieve directions. The user can mark favorite locations, which eliminates the need to input latitude and longitude each time to find a location. Google has also created and stored pictures from the street level for certain areas which can be accessed through Google Earth's layers list. Although the authors have not utilized this functionality with the described prototype, knowledge of the existence of these features within Google Earth may suggest options for further development.

2 Methodology

Although the Last.fm event-specific page offers detailed information about particular events, the site does not offer any way to view multiple events in a specific geographic area at a glance, or the popularity of multiple events simultaneously. This prototype represents the popularity of multiple events based on the attendance value reported by Last.fm by rendering a circle on the map centered on the geocoordinates of each event in the selected area with individual radii based on attendance, enabling the viewer to immediately observe the relative popularity of events within the area. A larger circle was chosen to be representative of an event with more people planning to attend, a smaller circle representing an event of lesser popularity. Events with none reporting as attending are marked with a pushpin indicating the location and details of the venue, but no circle is rendered. Contextual information associated with each event is accessed by the end-user clicking on the event pushpin, providing an intuitive interface that enables the end-user to browse and select upcoming events of interest – a benefit to the end-user that offers a clear improvement over the current lack of such an interface.

2.1 The Last.fm API

The strength of Last.fm and its mass database of artists, listeners and events becomes clear when using the Last.fm Web Services API, with the API available from http://www.last.fm/api offering methods for retrieving a number of different sets of data including information about specific artists, albums and songs, including specific user relationships and musical tastes. For this project we utilized the event tracking system, specifically the Geo.getEvents method of the API. The getEvents method returns a formatted list of the upcoming events (shows, concerts, etc.) within a specific geographical area. The Geo.getEvents web service is accessed through a standard URL including the method parameters. Available parameters include location, precise latitude and longitude, and radial distance out from the location center. Below is the getEvents URL call we used for this visualization:

```
http://ws.audioscrobbler.com/2.0/?method=geo.getev
ents&location=philadelphia&distance=2&api_key=b25b
959554ed76058ac220b7b2e0a026
```

Evident in the URL, the geo.getevents method requests information for events located within a radial distance of 2 kilometers of the location "philadelphia." The API key at the end validates the request with the Last.fm server to make sure it is authentic. This URL call then returns an extensible markup language (XML) document listing all of the upcoming events. The data was then manually transformed from the XML as returned by the API into the KML of a geobrowser visualization. Figure 5 includes a sample of the XML returned for a single event from the Last.fm API.

2.2 KML: Markup Language of Google Earth

KML, an abbreviation for "Keyhole Markup Language," is an XML-based language schema that was initially developed for Google Earth by Keyhole, Inc, a company that was subsequently acquired by Google. Use of KML allows the specification of features, placemarks, polygons and a wealth of other features along with the latitude and longitude of each encoded in a standard geobrowser format [6].

For this proof-of-concept prototype, the XML was

```
<event>
 <id>1092686</id>
 <title>(Folk Show)</title>
 <artists>
   <artist>UkeBox</artist>
   <artist>Jesse</artist>
   <artist>Mark Leonard</artist>
   <artist>Partysaurus Shreds!</artist>
   <headliner>UkeBox</headliner>
 </artists>
 <venue>
   <name>The Shore House</name>
   <location>
     <city>Philadelphia</city>
     <country>United States</country>
     <geo:point>
       <geo:lat>39.9350129603825</geo:lat>
       <geo:long>-75.157470703125</geo:long>
     </geo:point>
     <timezone>EST</timezone>
   </location>
   <url>http://www.last.fm/venue/9008647</url>
 </venue>
 <startDate>Mon, 08 Jun 2009</startDate>
 <startTime>18:00</startTime>
 <description>
   <div class="bbcode">Be there or be a circle!
      Hit up the bands for exact address.</div>
 </description>
 <image size="small">http://userserve-ak.last.fm/
                          serve/34/279397.jpg</image>
 <image size="medium">http://userserve-ak.last.fm/
                           serve/64/279397.jpg</image>
 <image size="large">http://userserve-ak.last.fm/
                          serve/126/279397.jpg</image>
 <attendance>1</attendance>
 <reviews>0</reviews>
 <tag>lastfm:event=1092686</tag>
 <url>http://www.last.fm/event/1092686</url>
</event>
```

Fig. 5. A snapshot of the XML returned by a call to the Last.fm API for the event specified by event id number 1092686.

```
<Placemark>
 <name>(Folk Show) at The Shore House</name>
 <description>
   <![CDATA[
   <img src="http://cdn.last.fm/flatness/badges/
                  lastfm_red_small.gif" /><br />
   <p><img src="http://userserve-ak.last.fm/
                    serve/126/279397.jpg" /></p>
   <h2>(Folk Show)</h2>
   With <b>Ukebox</b>, <b>Jesse</b>, <b>Mark
       Leonard</b> and <b>Partysaurus Shreds</b>
   <p><b>The Shore House</b><br /><br />
                                 Philadelphia</p>
   <p>Monday June 8th, 2009<br />
                      Currently Attending: 1</p>
   <p><a href="http://www.last.fm/event/9008647">
                     Last.fm Event Page</a><br />
   <div class="bbcode">Be there or be a circle!
        Hit up the bands for exact address.</div>
   <p><a href="http://music.myspace.com/index.cfm?
fuseaction=music.showDetails&amp;friendid=443847645
   ]]>
 </description>
       <Point>
        <coordinates>
          -75.157470703125,39.9350129603825,0
        </coordinates>
      </Point>
</Placemark>
```

Fig. 6. The XML returned by the call to the Last.fm API for the specific event id number of Fig. 1 parsed and rewritten as KML, the standard markup language used by Google Earth.

manually transformed into the KML schema, with the KML represented in Figure 6 showing the transformed XML of Figure 5; note the presence of the precise geographical location of the event in latitude and longitude. Relevant information about the show along with the description, venue location and other details extracted from the associated Last.fm event page have been added.

The markup has a similar look to the XML output returned by the Last.fm API call, but there are important differences; for example, the geo:point information from the XML document is now listed as coordinates in the KML, enabling Google Earth to pinpoint the exact location of the event on the map view. In addition, the details related to the show that in Figure 5 were contained in specific XML tags (artists, venue, date & time, etc.) have been combined into the single geo-recoded <description> tag of KML. The description tag is interpreted in Google Earth as a pop-up box that appears when the user clicks on the specific place mark. By implementing the CDATA operator within the description tag we are able to embed common HTML tags to format the data in a clear accessible manner, making the information readily accessible by the end-user.

3 Results

Figures 7-11 contain screen snapshots taken after opening the visualization in Google Earth viewer, and have been embedded as high-definition png files. The viewer application opens when the end-user double-clicks on the kml visualization file, which opens with the familiar Google Earth representation of the globe, then zooms in on the physical extent of the KML file: the 2km radius of the getEvent call surrounding the geo-coordinates associated with the area name “Philadelphia”. Table 1 contains the lookup table used throughout this implementation to map event popularity to encircling radius, where the use of a logarithmic response function and constants were chosen to maximize of event visibility, while maintaining a relatively tight association of the rendered circle to the event location on the visualization. Figure 7 contains a snapshot of the opening screen.

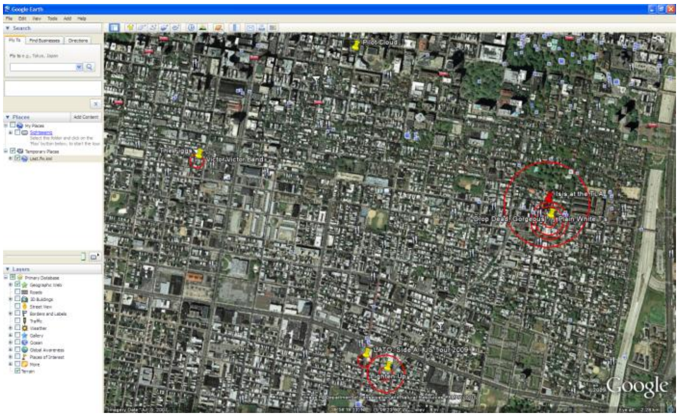


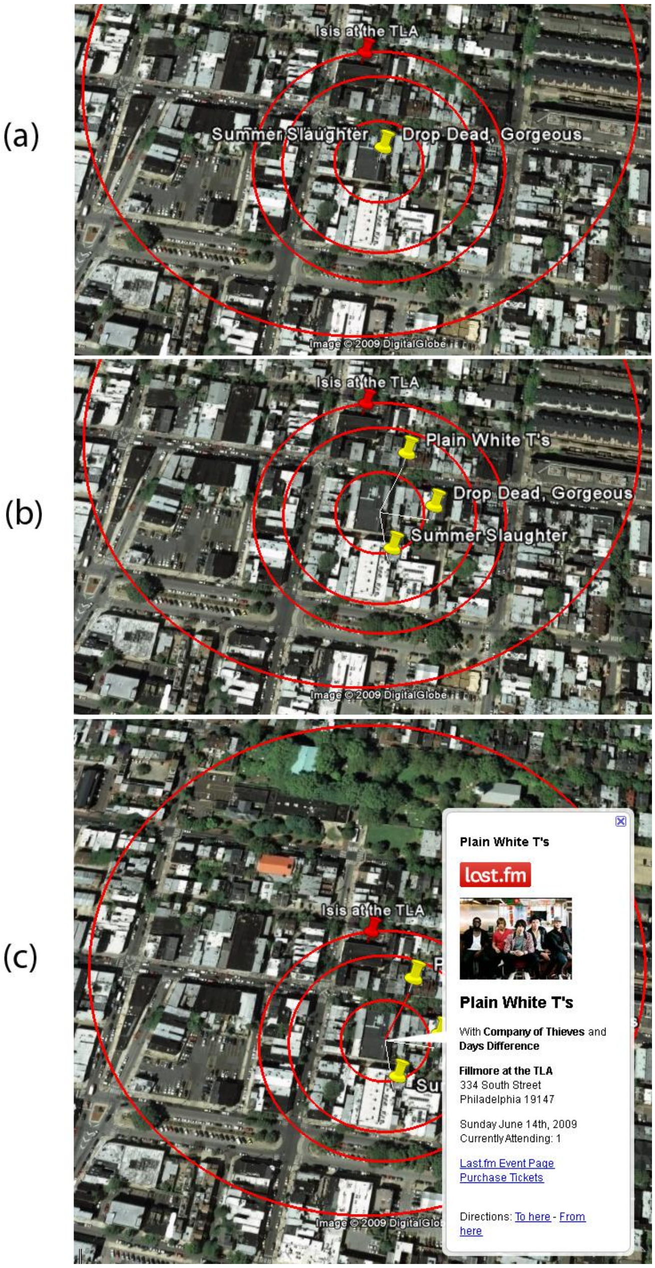
Fig. 7. The opening screen of the visualization of Last.fm events as visualized drawing from the getEvents call to the Last.fm API requesting events within a 2km radius of Philadelphia; note the existence of multiple events scheduled at a limited number of venues is clearly indicated as a larger number of red event circles than venue pushpins appear.

Figure 8 contains a montage of three screen shots, tracing the changes in the visualization as the user focuses on the large multiple-event venue on the center right-hand side of the initial extent of the viewer, and demonstrating the display choices made that enable a simplified view of multiple events located at a single venue. From the initial view in (a) followed by the separation of the single venue pushpin into three distinct events (b), closing with the information window that displays once the user has selected a particular event of interest by clicking on its pushpin icon (c); in the case of Figure 8, the event selected is associated with the headline event of the band "Plain White T's", appearing with two additional acts, at the venue "Fillmore at the TLA." The authors manually coded the html for each of the information windows con-

TABLE 1
MAPPING OF EVENT POPULARITY TO ENCIRCLING RADIUS

Event	Attending	Radius (m)
Isis	27	195
Pilot Cloud	0	0
Victor	0	0
Lighten Up	3	85
Summer Slaughter	3	85
(Folk Show)	1	30
Drop Dead, Gorgeous	2	65
The Figgs	1	30
Plain White T's	1	30
LATO	1	30

*A logarithmic function was chosen to map the size of the circle to event popularity based on the empirical maximum and minimum values present: 50*ln(# Attending) +30m. Events with no attendees are assigned a zero value; events with a non-zero number of attendees are mapped according to the relationship Radius = (50 • ln (Attending) + 30) m. Values were selected to maximize visibility and clarity of the depicting circles, with the logarithm ensuring that the circle remains small enough to remain visually associated with the event pushpin. m = meter.*



Plain White T's

Plain White T's

With Company of Thieves and Days Difference

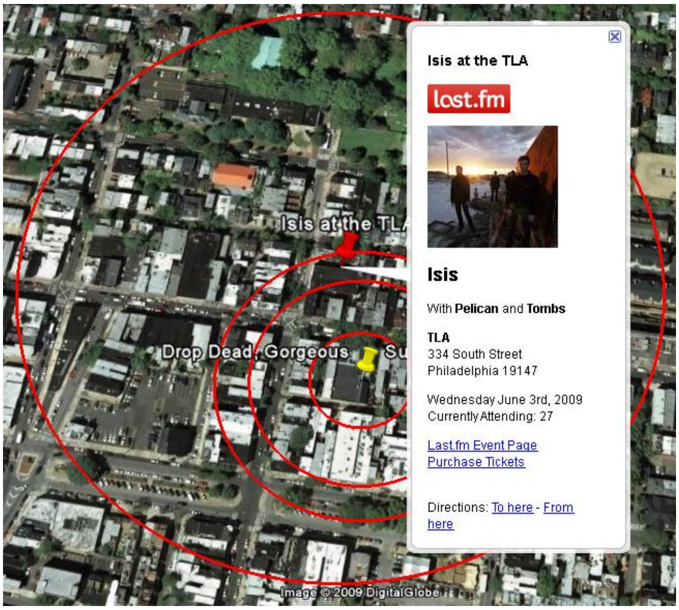
Fillmore at the TLA
334 South Street
Philadelphia 19147

Sunday June 14th, 2009
Currently Attending: 1

Last.fm Event Page
Purchase Tickets

Directions: To here - From here

Fig. 8. Zooming in on the multiple venue and events visualized on the center right of Fig. 7; note that in (a) there are four circles, while only three labels and two pushpins are drawn: to simplify the visualization, multiple events at a single venue are marked with a single pushpin, splitting into three separate events when the pushpin is clicked (b). Event details associated with each specific event were extracted from the return of the initial getEvents API call, and re-written as html embedded within a <![CDATA> tag in the <description> associated with the associated event in the KML file. The html event details render as a pop-up screen when the pushpin associated with an event of interest is clicked. Details include the neame of the headliner act, along with additional acts that will be present, date, venue address, number of attendees, along with separate hyperlinks to the Last.fm event page, a hyperlink to the ticket vendor (when applicable), along with hyperlinks to generate Google directions to and from the event.



Isis at the TLA

Isis

With **Pelican** and **Tombs**

TLA
334 South Street
Philadelphia 19147

Wednesday June 3rd, 2009
Currently Attending: 27

Last.fm Event Page
Purchase Tickets

Directions: To here - From here

Fig. 9. Rather than not showing events that have past, the choice was made to show events that have just completed; it is differentiated from upcoming events by coloring the pushpin red, while upcoming event pushpins share the color yellow. This particular event had 27 people indicate attendance, and is the event associated with the largest circle.

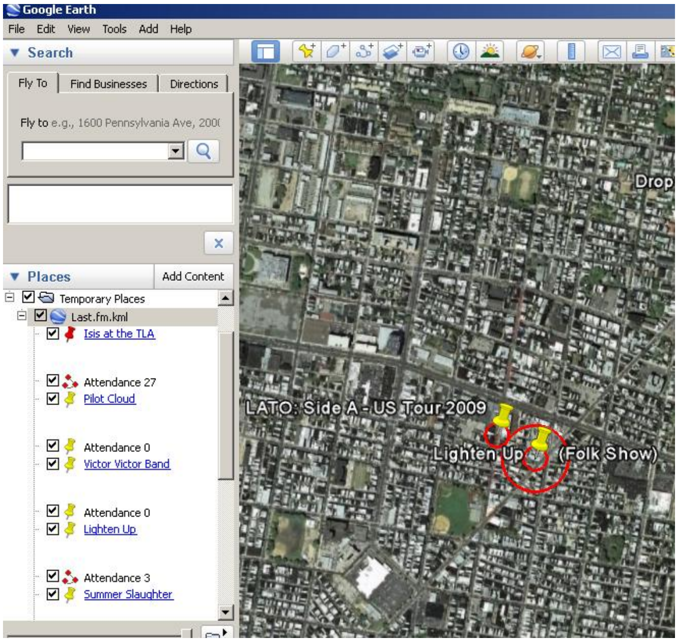


Fig. 11. In addition to the circle surrounding the event representative of the event popularity, the number attending has been embedded as a <name> tag within the <Placemark> tag associated with each event, and show up in the Places window on the left-hand side of the Google Earth application.

tained in the visualization, following the same selection rules for information to be displayed to maintain a cohesive display throughout. Based on our findings throughout, the authors anticipate that automation of such formatting and display would be eminently possible by following a similarly straightforward approach. As KML lacks a "circle" primitive, it should be mentioned that all popularity circles used in this prototype were manually constructed as polygons utilizing the <LineString> structure of the language.

Figures 9 and 10 demonstrate actively selected and programmed features of the visualization: Figure 9 shows the persistence of recently completed events of interest, albeit with a red pushpin rather than the default yellow pushpin for upcoming events, while Figure 10 displays the immediately apparent lack of a popularity circle; A quick glance at the information window that opens when the end-user clicks on the event pushpin reveals that there are currently no Last.fm users that have expressed plans to attend this particular event. The close-up shot in Figure 11 demonstrates the existence of the events geocoded and represented in the "Places" window of the Google Earth viewer; note the existence of the attendance numbers as part of the <Placemark> tagged specifically with each event, allowing the end-user to deselect the display of particular events as a matter of personal choice, by simply unchecking the event.

4 Discussion

The practical details associated with implementing the interface described required a considerable amount of trial and error, as it proved necessary to first try a possible visualization choice, such as shape, color, physical dimensions, etc, and then trial of the choice live in the Google Earth viewer. At all points throughout the development, our choices were informed by the active selection for a simplified view; an example of one such choice involved the initial choice to capitalize on the ability to tilt and pan the imagery in Google Earth, the authors initially implemented the popularity circle as a three-dimensional cylinder, by first drawing the circle at an altitude proportional to the event popularity, centered on the event coordinates, and extruding the shape to the ground. Although the effect was dramatic, the choice was made to opt for the simpler perception afforded by the use of a simple circle whose radius scaled with the event popularity. A further example is offered by the screen snapshots in this paper, where the outline of the circle appears to be thin – a reflection of the fact that the visually appealing thickness of the circle – when viewed in the Google Earth viewer – unfortunately renders less than satisfactorily in this print format. Nevertheless, the visualization is expected to be primarily used in the viewer, and the choice to maintain the optimal choices for that environment was made.

5 Conclusion

With this paper the authors have described the practical aspects of the development and initial testing os a proof-of-concept prototype user interface to the Last.fm community events data. The next steps should incorporate feedback from users of the Last.fm social networking community to determine the need for further development and improvement to the functionality of the visualization. Clearly, the steps the authors have identitfied in this document could be formalized in an application that could be implemented as a service; in such a case, it would be entirely possible for the events for an entire region or country to be returned in a single kml file, optimally compressed and returned as a kmz file. Alternatively, it may be possible to implement direct calls to the the Last.fm Web Services API, executing the calls directly from within the Google Earth viewer itself, parsing and re-displaying the returned events, perhaps as simply as with the use of an XSL style transformation embedded in a supporting application; although the authors did not directly explore that option, it would need added support from within the viewer interface, if only to implement the region query.

Although with this visualization the authors have barely scratched the surface of what is possible, clearly the visualization opens the door to what could prove to be a very useful tool in the lives of Last.fm users.

REFERENCES

[1] Lisle, R.J. 2006. "Google Earth: a new geological resource". *Geology Today*, vol. 22, no. 1., pp.29-33

[2] Archiving Early America. 2009. Retrieved from http://www.earlyamerica.com/earlyamerica/maps/bostonmap/bostonmap.jpeg

[3] United Airlines, 2009. Retrieved June 4, 2009, from http://content.united.com/ual/asset/UAL_NA_6_09.pdf

[4] N.A., 2009. Last.fm website, available at: http://www.last.fm/

[5] N.A., 2009. Last.fm Web Services API, available from: http://www.last.fm/api/intro

[6] Keyhole Markup Language (KML): Available at: http://earth.google.com/kml/