

Roland H. Schweitzer*
Weathertop Consulting, LLC, College Station, Texas

Steve Hankin, Ansley Manke
NOAA/Pacific Marine Environmental Laboratory, Seattle, Washington

Heather Koyuk, Karl Smith, Kevin O'Brien
JISAO, University of Washington, Seattle, Washington

1. INTRODUCTION

Many technology companies build and release software and services for public use. Java from Sun and now Oracle; Maps and Google Web Toolkit from Google; and Eclipse with contributions from many commercial enterprises are a few technologies with commercial roots that we have used successfully to build the Live Access Server. The Live Access Server is well-established Web-application software system for display and analysis of geo-science data sets. The LAS group has long been active in the development of tools to facilitate easy, Web-based model inter-comparison and other visualizations of earth science data. The LAS software, which can be downloaded and installed by anyone, gives data providers an easy way to establish services for their on-line data holdings so their users can make plots, create and download sub-sets in a variety of formats, and compare and analyze data. We are continuously evaluating and trying technology to see if we can use it to improve the services we provide around our core mission of making climate data available to scientists and the public. Google Fusion Tables is a new technology released for use by the public and we have begun the process of evaluating if and how we can use Fusion Tables and the services provided around them. This is a story both about technology: what it is and how it works, how we used it, the advantages it offers and where it falls short and it is a story about process: when and how do we integrate new technologies, when do we say no and how do we engage with the developers and the community to help it improve.

Initially, a Google Fusion Table looks like a cross between a database table and spreadsheet in the cloud. Data arranged in rows and columns can be uploaded into a table from local files in a variety of forms, from a Google document or using an API that

inserts data similarly to making inserts into an RDBMS. But, unlike the basic spreadsheets and RDBMS Fusion Tables layer additional services on top of the data to identify columns which contain location data and to provide map displays and a spreadsheet-like browser-based user interface. The data organization matches closely in-situ climate observation data for which we build display and analysis systems. We began our experiments using data available from the Observing System Monitoring Center (O'Brien, 2004, 2007, 2008) by extracting the sub-set from the RDBMS and manipulating it in a standard spreadsheet to create additional columns we needed to create the desired Google maps displays.

One of the first and most obvious advantages behind Google Fusion Tables in the tremendous amount of server-side processing that can be brought to bear on the data to create sub-sets, perform calculations and to build interactive Google-maps compatible overlay tiles with representations of the data which can displayed on a map in the browser and can interactively link back to the underlying table row.

Of course, all of that processing is not free and there are limitations both on the amount of data that can be uploaded and stored and the number of different layers created and displayed on a map. And while designed to be used with general location-based data collections in many disciplines, some features which seem obvious for Earth-science data are lacking or difficult to archive.

The technology is new so learning how to use it and discovering what it could and could not do involved not only reading the available documentation it involved engaging with the community. One of the criteria for evaluating a new technology is the how active and engaged the community is with developers and other users.

Our initial experimental with the technology was very promising. Uploading, sharing and displaying were easy. And when combined with some basic JavaScript programming we were able quickly build some high quality, interactive map displays of our data sub-set. However, there are still many questions to

*Corresponding author address: Roland H. Schweitzer, Weathertop Consulting, LLC, 2802 Cimarron Ct, College Station, TX 77845. E-mail: weathertop.consulting@gmail.com

the map.

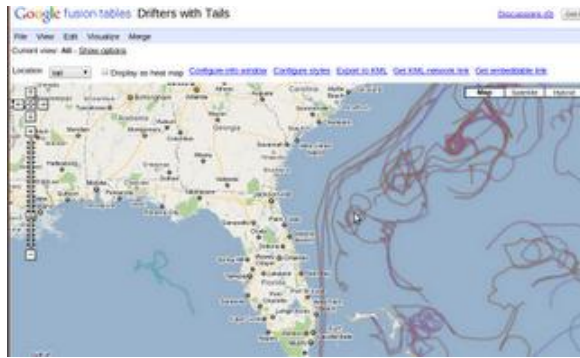


Figure 3 The individual drifters shown as line segments.

4. USING THE MAPS API AND THE FUSION TABLE LAYER

In the user interface provided, you can only map one "location" at a time, so you cannot simultaneously show the KML fragments that describe the tail and the dots that show the sample locations. Even if you could it would be a really big mess with the dots all over the line segments. What we want to present is the drifter's current location and its tail. I used a spreadsheet function to create a column which was blank everywhere except the final location of the platform. This function `=IF(A3=A4,"",I3)` says if it's the same as the next one, blank, if the next one is different mark the final position. But, when I plot that column in the UI I only get a few of the final locations. I don't know if this is a bug making the map tiles, but I found that if I put the final positions in a separate table I could map them using the Maps API and get the desired effect.

Using the SQL API you can request data from your table and using the V3 Maps API you can get Google to create and deliver map tiles with plots of the data that overlay on the map. And not only that, the layers are active and accept clicks which will pop up an info window. And not only that you can register a click listener on the layer which gets the location clicked, the table row at that location and an info window object you can populate with HTML.

And since the row contains the ID of the platform, it is possible to place a button in the HTML that will do a query that hides all of the other platforms and displays only the tail and final position of the platform selected by the click.

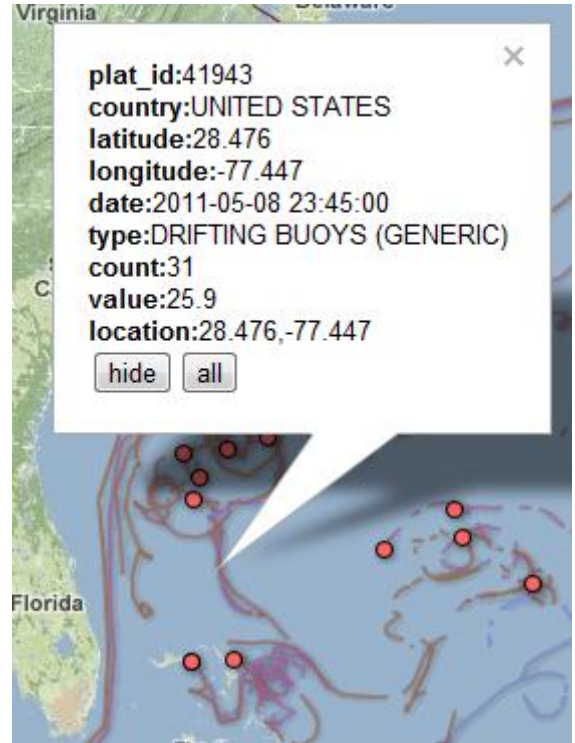


Figure 4 Clicking on a line give the data for that time and shows a button to hide the other platforms.

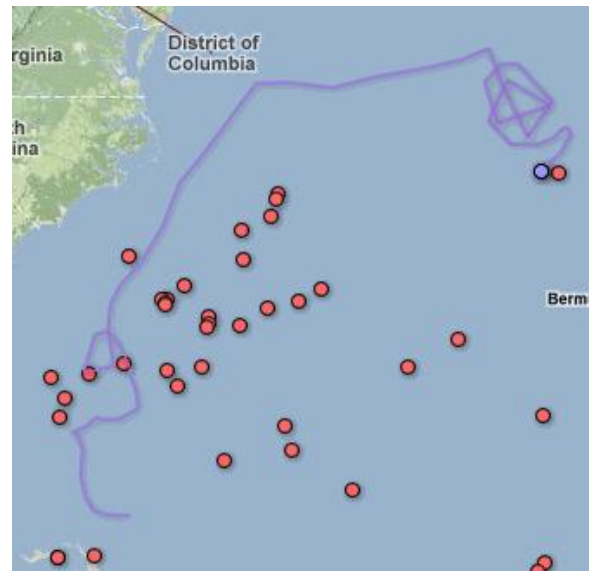


Figure 5 Pushing the hide button shows only the selected drifter.

7. REFERENCES

O'Brien, K., K. McHugh, G. Vecchi, E. Harrison, S. Hankin, and A. Manke (2004): The Observing System Monitoring Center: A Tool for Evaluation of the Global

Ocean Observing System. In Proceedings of the 20th International Conference on Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology, 2004 AMS Annual Meeting, Seattle, WA, 12–15 January 2004, paper P1.35.

O'Brien, K.M., S. Hankin, R. Schweitzer, K. Kern, B. Smith, T. Habermann, and N. Auerbach (2007): An introduction to the Observing System Monitoring Center. In Proceedings of the 23rd International Conference on Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology, 87th AMS Annual Meeting, San Antonio, TX, 14–18 January 2007, Paper 2B.5

O'Brien, Kevin, S. Hankin, R. Schweitzer, K. Kern, M. Little, T. Habermann, N. Auerbach, J. Cartwright, J. LaRocque (2008): MONITORING AND ANALYZING THE GLOBAL OCEAN OBSERVING SYSTEM WITH THE OBSERVING SYSTEM MONITORING CENTER in Proceedings of the 24th International Conference on Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology, 88th AMS Annual Meeting, New Orleans, LA, 20–24 January 2008, Paper J1.6