Jul 12th, 9:50 AM - 10:10 AM

Time Series Analyst: Interactive Online Visualization of Standards Based Environmental Time Series Data

Jeffery S. Horsburgh  
*Utah State University,* jeff.horsburgh@usu.edu

Amber Spackman Jones  
*Utah State University,* amber.jones@usu.edu

Maurier Ramírez  
*Utah State University,* mauriel.ramirez@gmail.com

Juan Caraballo  
*Utah State University,* juan.caraballo17@gmail.com

Follow this and additional works at: [https://scholarsarchive.byu.edu/iemssconference](https://scholarsarchive.byu.edu/iemssconference)

Part of the Civil Engineering Commons, Data Storage Systems Commons, Environmental Engineering Commons, Hydraulic Engineering Commons, and the Other Civil and Environmental Engineering Commons

Horsburgh, Jeffery S.; Jones, Amber Spackman; Ramírez, Maurier; and Caraballo, Juan, "Time Series Analyst: Interactive Online Visualization of Standards Based Environmental Time Series Data" (2016). *International Congress on Environmental Modelling and Software.* 12.  
[https://scholarsarchive.byu.edu/iemssconference/2016/Stream-A/12](https://scholarsarchive.byu.edu/iemssconference/2016/Stream-A/12)

This Event is brought to you for free and open access by the Civil and Environmental Engineering at BYU ScholarsArchive. It has been accepted for inclusion in International Congress on Environmental Modelling and Software by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.
Time Series Analyst: Interactive Online Visualization of Standards Based Environmental Time Series Data

Jeffery S. Horsburgh, Amber Spackman Jones, Maurier Ramírez, Juan Caraballo
Department of Civil and Environmental Engineering and Utah Water Research Laboratory, Utah State University, 8200 Old Main Hill, Logan, Utah, 84322-8200, USA (jeff.horsburgh@usu.edu, amber.jones@usu.edu, mauriel.ramirez@gmail.com, juan.caraballo17@gmail.com)

Abstract: Visualization is a common need of many researchers, organizations, and projects that collect and use environmental sensor data. Web-based tools can provide screening-level visualization and data analysis functionality for users with a range of technical expertise. However, it is difficult to integrate data from multiple sources due to syntactic and semantic differences in data formats and different data delivery mechanisms that are not always interoperable. We developed an open-source time series visualization tool called the Time Series Analyst (TSA) that integrates data from multiple sources using web service interfaces and a standardized data encoding format. The TSA provides multiple types of visualization presented through a web browser. Important functionality includes a map-based interface and faceted filters to narrow the selection and display of a large number of data series. Data are retrieved using web services that comply with standards established by the Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI) Hydrologic Information System (HIS), and any time series dataset published using the CUAHSI HIS can be visualized using TSA. Finally, we present a case study demonstrating how TSA can be used to visualize data from a large-scale environmental sensor network.

Keywords: Time series; Sensor data; Web Services; Visualization; Interoperability

1 INTRODUCTION

It is increasingly common for research groups, organizations, and agencies to collect time series data using in situ environmental sensors (Hart and Martinez, 2006; Rundel et al., 2009; Muste et al., 2013). These data can be challenging to manage, especially as the number of sites, variables, and the time period over which observations are collected increases. One major challenge that many projects and data managers face is providing convenient access to visualize and download observational data for users with different technical skill levels (Horsburgh et al., 2011; Muste et al., 2013; Demir and Krajewski, 2013; Mason et al., 2014). Additionally, it is challenging to present data in the context of datasets collected by other groups and organizations in a consistent way.

In this paper, we describe a software application called the Time Series Analyst (TSA) that we developed for presenting interactive visualizations of environmental time series data on the Internet. The TSA uses faceted filtering of series to narrow data series for selection, and data collection locations are also presented on an interactive map to provide spatial context. Users can plot data using time series, box-and-whisker, and histogram plots. The web application retrieves data from web services published using the Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI) Hydrologic Information System (HIS) (Horsburgh et al., 2009; Tarboton et al., 2009). The CUAHSI HIS includes standards-based web services for publishing hydrologic time series on the Internet (Zaslavsky et al., 2007). The TSA is capable of retrieving and visualizing any time series for which CUAHSI HIS web services have been deployed. A growing list of published web services is available at http://hiscentral.cuahsi.org. The TSA described here is a complete rewrite of an older version released as part of the CUAHSI HIS in 2011. We added a dynamic map, support for deployment on multiple platforms, and interactive visualizations using JavaScript.
2 SOFTWARE IMPLEMENTATION

The TSA was developed to present visualizations of environmental time series data from multiple sources using simple, robust, and interactive methods. The following requirements drove our design:

1. A web browser-based graphical user interface (GUI) for use on any computing platform
2. Simple filtering of available time series datasets via search facets and a map-based interface to select time series for visualization
3. Dynamic and interactive plotting with zooming and panning through time
4. Retrieval of data to be visualized from standards-based web services deployed by multiple data sources
5. Export of selected time series datasets to a simple, text-based format
6. Support for deployment on both Windows and Linux server platforms

2.1 Overall Architecture

The TSA is a web application with a web browser-based GUI. It was developed using HTML, CSS, the Django Python web framework (https://www.djangoproject.com), and several JavaScript libraries (see Software Availability), which can all be deployed on multiple platforms. Figure 1 shows the overall architecture of the TSA. Time series datasets are retrieved for visualization via CUAHSI HIS WaterOneFlow web services from a server on which they are stored. Any time series dataset published using CUAHSI HIS web services can be visualized. A catalog database stores the list of time series available for retrieval and visualization (a defined subset of available CUAHSI HIS time series). When the GUI is loaded into a web browser, it queries a web service that accesses the catalog to retrieve and display the list of available time series. In the following sections we describe each of these components, including their main functionality and the technologies used to develop them.

2.2 Time Series Catalog and Data Retrieval

The TSA was designed to retrieve time series datasets for visualization from any CUAHSI HIS WaterOneFlow Version 1.1 web service. This provides flexibility for deployment as time series published by a particular group can easily be combined with those available from other WaterOneFlow web services (e.g., those available for the United States Geological Survey’s (USGS) National Water Information System (NWIS)). We also wanted to have granular control over which time series were available for visualization. To accomplish this, we created a catalog database containing a single table for storing the list of time series datasets available for visualization, with the metadata necessary to provide a full description of each time series and to enable faceted browsing using metadata elements as facets. Attributes in the time series catalog table are based on the CUAHSI HIS Observations Data Model (ODM) (Horsburgh et al., 2008). One column in the catalog table contains a URL that can be used to programatically retrieve the time series dataset from its WaterOneFlow web service. The catalog must be generated prior to deployment of the TSA. Our current deployment of the TSA uses Microsoft SQL Server for the catalog, but, with minor modifications to configuration files when deploying the application, MySQL, PostgreSQL, or SQLite could be used. Instructions for generating the catalog database are available in the TSA GitHub repository. The contents of the catalog database are exposed and can be queried through a catalog web service that serializes the list of available time series as JSON. The catalog service was implemented using the Tastypie web service API framework for Django (http://tastypieapi.org).

2.3 Graphical User Interface

The TSA GUI consists of a faceted browsing panel linked to three tabs containing the visualization capabilities. The “Map” tab plots the locations at which time series have been collected, the “Datasets” tab lists the available time series and their attributes, and the “Visualization” tab presents the data visualizations. These four components are dynamically linked. Selections on the faceted browsing panel determine which data collection locations are shown on the Map tab and which time series are shown in the Datasets tab. Similarly, selections on the Datasets tab determine which time series are shown in the Visualization tab. Each component is described in more detail in the following sections.
2.3.1 Faceted Browsing Panel

When the TSA loads in the browser, the time series catalog web service is queried to retrieve the list of available time series datasets. Values for time series attributes from the catalog are loaded as categories into the faceted browsing panel (left column panel in Figure 2), which was implemented using JavaScript. User-selections by clicking the check boxes within the search facets are used to filter the list of data collection sites shown on the Map tab and the time series datasets shown on the Datasets tab. Selections within a facet (e.g., selecting a particular data collection site) narrow the options available in the other facets, limiting them to the subset of values that is consistent with previous selections (e.g., the list of variables that can be selected is automatically limited to those available at the selected site). The list of available search facets (the categories such as Network, Site, Variable, etc. in the facet panel) is customizable when the TSA is deployed. Facets can be selected from the existing time series attributes in the catalog, and new facets can be used by adding them as attributes to the catalog database.

2.3.1 Map Visualization Tab

The Map tab (Figure 2) uses the Google Maps JavaScript application programmer interface (API) (https://developers.google.com/maps/web/). Locations at which time series datasets were collected are shown on the map with markers that vary by style (indicating the site type – e.g., aquatic or climate) and by color (indicating user-defined site groupings – e.g., USGS daily streamflow sites or local research network). Clicking on the marker for an individual site opens a pop up bubble listing attributes of the site and providing a button to load the time series datasets for that site into the Datasets tab. Furthermore, when a site is selected on the map tab, options available in the faceted browsing panel are filtered to display only attributes available at that data collection location. Using JavaScript, we incorporated clustering of the site markers so that when users zoom out, the map does not become cluttered if there is a large number of site markers.
The Datasets tab lists the time series datasets that are available for visualization (Figure 3). It responds dynamically to selections made on the faceted browsing panel and selections made on the Map tab to display the list of time series meeting user specified filtering criteria. The Datasets tab corresponds to the catalog, where each row in the table represents a single time series, and the columns in the table provide metadata. The table, which was implemented using the DataTable JavaScript library (http://www.datatables.net), can be customized to show or hide metadata columns according to a user’s preference. Multiple datasets can be selected for visualization by clicking a check box in the left-most column in the table. Clicking on the row in the table displays a pop-up window with a full metadata description of the time series. On the pop-up there are buttons to export the time series to a file or to add the time series to the current plot. Additional functionality includes a keyword search for filtering displayed time series, an option to show only selected or show all time series, and a button for exporting a zipped file containing multiple selected time series formatted as ASCII text files with a metadata header and the times series data as comma separated values.

2.3.3 Visualization Tab

The Visualization tab consists of a plot panel with additional panels for zooming/panning control, setting plot options, a legend, and summary statistics (Figure 4). All of the time series selected for visualization on the Datasets tab are shown in the plot panel and the legend. The legend panel also allows users to toggle visibility for each time series in the plot. Summary statistics are calculated and displayed for a single time series that is selected in the legend panel. In the plot options panel, users can select the time period for which the visualization should be drawn (i.e., full time range, the past month, the past week) and the type of visualization to draw (i.e., time series, histogram, or box-and-whisker). In all plot types, each individual time series is visualized using a different color. For time series plots, time series of different units are plotted using different y-axes, and the corresponding
The y-axis for each time series is color coded to match the plotted time series line. Finally, the zooming/panning panel enables users to dynamically zoom in on the time series plot and then drag the selection to pan the plot to different time periods. The plots on the Visualization tab were all implemented using the D3.js JavaScript library (http://d3js.org).

2.4 URL Patterns for Launching the TSA

The TSA is capable of being launched in a particular state – i.e., with a particular time series selected and plotted on the Visualization tab when the application loads in the web browser. This is useful where the TSA is used as a companion to a separate website that includes links to datasets available for visualization using the TSA. The general URL pattern for launching the TSA is as follows:

http://{DomainName}/tsa/?sitecode={SiteCode}&variablecode={VariableCode}&qualitycontrollevelcode={QualityControlLevel}&view={SelectedView}&plot={PlotTrueFalse}

Where:

{DomainName} = domain name under which the TSA is deployed
{SiteCode} = unique code identifying the site at which the data were collected
{VariableCode} = unique code identifying the measured variable
{QualityControlLevel} = identifier for the version of the data to plot (raw or quality controlled)
{SelectedView} = name of the TSA tab to show when the application launches
{PlotTrueFalse} = Boolean parameter indicating whether the plot should be drawn
3 CASE STUDY

The TSA was conceptualized and implemented as a visualization tool for the innovative Urban Transitions and Arid-region Hydrosustainability (iUTAH) Gradients Along Mountain to Urban Gradients (GAMUT) observatory of environmental monitoring sites. iUTAH (http://iutahepscor.org) is an interdisciplinary project with a broad audience needing access to the time series data. Users have varying levels of technical expertise, including researchers from different domains, municipality and local agency partners, educators, technical support staff, and the general public. Implementing the TSA for the GAMUT data has provided an easily accessible interface to this wide range of users.

The GAMUT network consists of in situ aquatic, climate, and terrestrial sensors deployed at sites within three watersheds (Logan River, Red Butte Creek, Provo River) in northern Utah, USA. The filtered browsing of data series in the TSA facilitates access to data from the large number of sites and variables (40 sites and 135 variables at the time of writing) in the GAMUT network. The CUAHSI HIS software stack (Horsburgh et al., 2010) was deployed to store, manage, and publish the GAMUT data. The TSA catalog database was initially generated by querying attributes of the GAMUT time series from the ODM databases in which the data are stored, including construction of the web service URL. These queries are automatically run daily as scheduled stored procedures to update the catalog database as new data are collected. The catalog database is exposed via the catalog web service, which always delivers the most recent metadata from the catalog database.

We also added time series from USGS NWIS sites within the GAMUT watersheds to the TSA catalog database to enable simultaneous visualization with GAMUT datasets via the TSA. The USGS Sites web service (http://waterservices.usgs.gov/rest/Site-Service.html) was used to find available time series for USGS NWIS data sources within the GAMUT watersheds for which WaterOneFlow web services are available (i.e., Daily Values, Instantaneous Values, and Groundwater Levels). Queries
were developed to add the associated records and their web service URLs to the TSA catalog database. With those sources in the catalog database, the USGS data are equally accessible for faceted browsing, mapping, and visualization in the TSA alongside the GAMUT data.

In the implementation of the TSA for the GAMUT observatory (http://data.iutahepscor.org/tsa/), search facets include Network (e.g., USGS Daily Values, Red Butte Creek), Site (e.g., Red Butte Creek at Foothill Drive), Variable Category (e.g., climate, soil, water quality), Variable Name (e.g., air temperature, pH, snow depth), and Quality Control Level (e.g., raw, quality controlled). We also implemented a custom facet called ‘Variable Level’ to specify whether the data series are ‘Common’, referring to a series in which data users might have interest as opposed to ‘Uncommon’ that are troubleshooting variables relevant to technicians. When the TSA launches, its default is to have the ‘Common’ facet selected to provide a narrowed list of variables to users, but the option for the full list is still available. We implemented webpages for each of the GAMUT sites with spark line plots of variables showing data for the past 24 hours (e.g., http://data.iutahepscor.org/mdf/river_info/UTAH_Logan_OD/LR_WaterLab_AA/). Clicking a sparkline launches the TSA with the full time series visualized as described in Section 2.4. The URL launched by clicking on a sparkline follows the pattern shown in Section 2.4 (e.g., http://data.iutahepscor.org/tsa/?sitecode=LR_WaterLab_AA&variablecode=SpCond&qualitycontrollevelcode=0&view=visualization&plot=true).

4 DISCUSSION

The combination of the map-based display of monitoring locations, the faceted browsing interface, and the visualizations available in the TSA provides a mechanism for filtering and providing quick, screening-level visualizations for the large number of time series datasets in our case study. No specialized software or expertise is required to use the tool, which significantly lowers the barrier for accessing the data. Exporting selected datasets in text format for more sophisticated analyses in separate data analysis software is straightforward.

In our experience, the D3.js visualization tool provides very fast and interactive visualizations, even when the number of observations is in excess of 100,000. The bottleneck for our current implementation is in retrieving the time series data in WaterML format from the CUAHSI HIS WaterOneFlow web services. The bulk of the loading time for the Visualization tab is in waiting to retrieve the data from the web service. Once the data are retrieved, they are parsed relatively quickly. One potential solution may be to add a lightweight GetValues method to the CUAHSI HIS WaterOneFlow web services that returns the data as JSON with much less overhead than the current full XML WaterML response. So, while standards-based web services make the data from multiple sources interoperable and reusable within a single GUI that consistently presents data from multiple sources to users; there is a definite tradeoff in performance (i.e., which manifests itself in the time it takes to present the interactive plots to the user) related to retrieving the data on demand from the web services in WaterML format.

The TSA can easily be adapted for reuse. The catalog database provides very granular control over which time series are made available for visualization. This fine level of control can be used to combine time series for an experimental watershed or site with a subset of time series available from national scale web services (e.g., the USGS NWIS system for the whole country) as we did for iUTAH. Furthermore, the ability to launch the TSA in a particular state (e.g., with a particular time series selected and plotted) makes it useful for pairing with other websites that provide listings of available data or data collection locations. Indeed, we anticipate that the TSA may be useful for many different organizations that need to provide public visualization and data access capabilities for time series datasets.

SOFTWARE AVAILABILITY

The Time Series Analyst Software is available for download from the GitHub repository at https://github.com/UCHIC/WEBTSA. A full list of dependencies is provided with the source code and documentation in the repository. Software and code are released under the New Berkeley Software Distribution (BSD) License.
ACKNOWLEDGMENTS

This work was supported by the National Science Foundation under EPSCoR grant IIA 1208732 awarded to Utah State University as part of the State of Utah EPSCoR Research Infrastructure Improvement Award. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES


