



Open source software for visualization and quality control of continuous hydrologic and water quality sensor data



Jeffery S. Horsburgh ^{a,*}, Stephanie L. Reeder ^b, Amber Spackman Jones ^b, Jacob Meline ^b

^a Department of Civil and Environmental Engineering, Utah Water Research Laboratory, Utah State University, 8200 Old Main Hill, Logan, UT 84322-8200, USA

^b Utah Water Research Laboratory, Utah State University, Logan, UT, USA

ARTICLE INFO

Article history:

Received 9 January 2015

Received in revised form

10 March 2015

Accepted 9 April 2015

Available online

Keywords:

Data management

Open source software

Sensor data

Python

Hydrologic observations

Quality control

ABSTRACT

It is common for *in situ* hydrologic and water quality data to be collected at high frequencies and for extended durations. These data streams, which may also be collected across many monitoring sites require infrastructure for data storage and management. The Observations Data Model (ODM), which is part of the Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI) Hydrologic Information System (HIS), was developed as a standard data model in which to organize, store, and describe point observations data. In this paper we describe ODM Tools Python, an open source software application that allows users to query and export, visualize, and perform quality control post processing on time series of environmental observations data stored in an ODM database using automated Python scripting that records the corrections and adjustments made to data series in the quality control process and ensures data editing steps are traceable and reproducible.

© 2015 Elsevier Ltd. All rights reserved.

Software availability

Name of software: ODM Tools Python

Developers: Jeffery S. Horsburgh, Stephanie L. Reeder, Amber Spackman Jones, Jacob Meline, and James Patton

Contact: jeff.horsburgh@usu.edu

Year first available: 2014

Hardware required: A personal computer

Software required: Microsoft Windows, Mac OSX, or Linux operating system

Software availability: All source code, installers, example ODM databases, and documentation for the ODM Tools Python software application can be accessed at <https://github.com/UChIC/ODMToolsPython>.

Cost: Free. Software and source code are released under the New Berkeley Software Distribution (BSD) License, which allows for liberal reuse of the software and code.

1. Introduction

Environmental monitoring with *in situ* environmental sensors presents many challenges for data management, particularly for large-scale networks consisting of multiple sites, sensors, and personnel. Over the past decade, there has been a drastic increase in the use of automated data collection in scientific research. The high frequency, extended duration, and spatial distribution of data collection efforts require cyberinfrastructure to support and facilitate research using sensor data streams. Researchers and practitioners need tools for data import and storage as well as data access and management. In addition to addressing the challenges presented by managing the sheer quantity of data, monitoring network managers need practices to ensure high data quality, including standard procedures and software tools for data post processing and quality control.

In this paper we describe a workflow for scripted quality control editing of continuous, *in situ* time series datasets and the architecture and functionality of an open source software tool called ODM Tools Python that implements this workflow. ODM Tools Python enables users to query and export, visualize, and edit time

* Corresponding author. Tel.: +1 435 797 2946.

E-mail address: jeff.horsburgh@usu.edu (J.S. Horsburgh).

series observations stored in an Observations Data Model (ODM) database (Horsburgh et al., 2008). ODM was developed as a standard data model in which to organize, store, and describe point observations (e.g., observations made at fixed point monitoring sites such as streamflow, water quality, and weather monitoring stations) with sufficient metadata for observations to be unambiguously interpreted by multiple users. ODM is implemented in relational database software to permit flexibility in querying and data retrieval. ODM Tools Python is a modernized and advanced version of the original ODM Tools software, which, along with ODM, was developed as part of the HydroServer software stack (Horsburgh et al., 2010) within the Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI) Hydrologic Information System (HIS) (Horsburgh et al., 2009; Tarboton et al., 2009).

Previous versions of ODM Tools (Horsburgh et al., 2011) were developed in Microsoft Visual Studio .NET and only supported ODM databases implemented in Microsoft SQL Server, limiting deployment to Microsoft Windows-based computers. Functionality consisted of exporting data series and associated metadata, plotting and summary of single data series, generation of derivative data series, and editing data series using a set of simple tools implemented within the ODM Tools graphical user interface (GUI). However, there were no capabilities for recording or storing the sequence of edits made to a data series in the quality control process. Furthermore, there were not adequate capabilities to support the full quality control workflow, especially for long time series that were difficult to process in a single editing session or time series continually being updated as new data were collected. These were major limitations that affected the usefulness of ODM Tools where recording the provenance of data edits was desired or required. Irrespective of the tool being used to perform quality control, it remains common for some scientists to perform these steps at the same time as data visualization and analysis without preserving the provenance of the quality control process in a way that it is accessible to others or for future reference. We sought to address this deficiency with the software described in this paper.

The new, Python-based version of ODM Tools described here adds a modernized GUI with dockable components, multiple platform support (Windows, Linux, and Mac), support for multiple relational database management systems (RDBMS) (Microsoft SQL Server, PostgreSQL, and MySQL), enhanced plotting and visualization, and automated scripting of quality control edits performed on data series through an integrated Python script editor and console. Additional improvements include enhanced queries for data selection and export, interactive data selection directly on a time series plot, additional options for how edits to data series are versioned and saved, and the ability to plot multiple data series simultaneously with various plot types. The contribution of this paper is in the description the quality control editing workflow for continuous sensor data streams and in the architecture of the ODM Tools Python software that implements the workflow and enables the features listed above. We anticipate that the functionality that we have developed in ODM Tools Python will be widely applicable to managers of continuous sensor data and that the architecture, software development approach, and deployment approach that we have used for the ODM Tools Python software will be informative for researchers developing similar tools.

In Section 2 we provide further background on quality control for hydrologic and water quality sensor data streams and associated challenges. Section 3 describes our workflow for supporting the quality control editing process. Section 4 describes our approach to address these challenges and the architecture and software implementation for ODM Tools Python. Section 5 describes the deployment approach we selected for the ODM Tools Python

software, and Section 6 provides an example of using ODM Tools Python for quality control editing and post processing of a continuous water quality time series. Finally, we conclude with a summary of the research results.

2. Background

2.1. Quality control of sensor data

Errors in continuous environmental datasets primarily occur from fouling of sensors and from sensor calibration shift, although anomalies and erroneous data values can occur for many other reasons, including failures of sensors, recorders, transmission systems, or unforeseen environmental conditions that adversely affect sensor readings (Wagner et al., 2006). Producing high quality, continuous data streams from raw sensor output requires application of data corrections to mitigate for instrument calibration drift, fouling, and other errors (Mourad and Bertrand-Krajewski, 2002; Campbell et al., 2013). Fig. 1 shows common types of errors in raw sensor data streams and illustrates the associated types of corrections needed in the quality control process. These are also listed in Table 1, which provides a more extensive list of common error types along with examples and description of the most common procedures for correction. The errors listed in Table 1 are illustrative of those commonly encountered within raw environmental sensor datasets, and the software described in this paper was developed to advance data from raw to quality controlled versions using the corrections described in Table 1.

Correction procedures listed in Table 1 include simple methods such as: deletion of erroneous values, insertion of individual values to fill brief no-data gaps, and interpolation of erroneous values using prior and subsequent values that are known to be good. More sophisticated corrections include variable data corrections applied to correct for sensor drift and fouling. It is beyond the scope of this paper to describe the circumstances under which each of these corrections should be applied. Rather, we describe a workflow and software that enable application of any of these corrections according to the judgment of the data analyst in a way that the provenance of the changes can be saved. Wagner et al. (2006) provide excellent guidance and discussion on which correction procedures to use for various data collection situations encountered in hydrologic and water quality monitoring. Similarly, Fiebrich et al. (2010) review checks and correction procedures for common meteorological variables.

Another important distinction is that of “automated” quality control versus “interactive” quality control editing performed by an analyst. Most automated quality control procedures are focused on automatically flagging raw data values that do not meet one of several plausibility tests (e.g., Sheldon, 2008; Lerner et al., 2011; Taylor and Loescher, 2013). These include programmatic checks aimed at identifying instances of the types of errors described in Table 1 in raw data streams. Algorithms for identifying these errors range from simple range checks to more complicated techniques using statistical models or machine learning methods (e.g., Moatar et al., 2001; Hill et al., 2009; Fiebrich et al., 2010; White et al., 2010; Dereszynski and Dietterich, 2012). While these techniques can be quite good at identifying and flagging potentially erroneous values, they do not excel at choosing and applying appropriate data corrections to resolve the error for which the data were flagged, steps that generally require the attention and expertise of field or data technicians (Fiebrich et al., 2010; White et al., 2010). This paper, the workflow we present, and the ODM Tools Python software focus on this secondary level of quality control (or post processing) and provide a set of tools for applying the common data corrections listed in Table 1.

Download English Version:

<https://daneshyari.com/en/article/6963105>

Download Persian Version:

<https://daneshyari.com/article/6963105>

[Daneshyari.com](https://daneshyari.com)