Critical Zone Observatories and Sensor Repositories

Stephen Wilson
School of Chemistry, University of Southampton, Southampton, UK sw1703@soton.ac.uk

Jeremy Frey
School of Chemistry, University of Southampton, Southampton, UK j.g.frey@soton.ac.uk

10th July 2009

1 Introduction

The Shale Hills critical zone laboratory in Pennsylvania was one of three critical zones funded by the U.S National Science Foundation[1]. The goal of these areas was to study the complex processes occurring on the Earth's surface, including research into hydrology, geomorphology and biogeochemical systems. Acknowledging that this is a multi-discipline problem the funding promoted interdisciplinary research within these areas. To allow sharing of this data between researchers a standardised approach of storage the data is required. The volume of data and need for automation means that the data must also be accessible through machine to machine interactions.

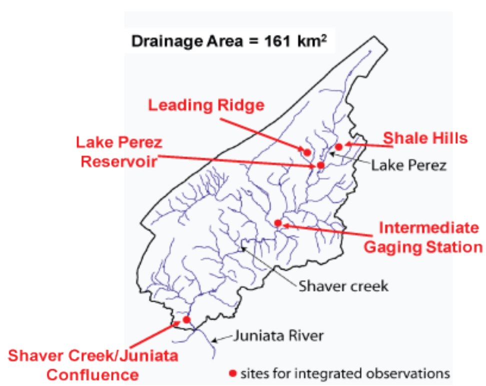


Figure 1: The Shale Hills Critical Zone - src: http://www.rthnet.psu.edu/

Capture and storage of sensor data has been possible for a long time, this data has typically been stored locally using a custom format known only to the researchers handling the data. These custom data models have made it difficult to share data with other research groups and to link into commercial software. The use of XML has enabled the development of standard data models such as SensorML[2], Observation & Measurement and KML (Keyhole Markup Language) which have promoted machine to machine communication and the sharing of data. The development of API protocols have also made it easier to make this data available to a wider community.

In this work we propose the use of an Open Geospatial Consortium (OGC)[7] standard sensor repository model, the Sensor Observation Service. The standard describes an API for managing deployed sensors and retrieving the data from these sensor (either in-situ or dynamic sensors). As this is a global standard other systems will be able to interact and make use of the data. Additionally, the trust in the data can be improved as the meta data relating to a given measurement and the sensor making the measurement will be available (such as sensitivities and operational ranges). There are three core operations the SOS must offer; getCapabilities (retrieves all operations and allowed values the SOS offers), describeSensor (returns the metadata of a given sensor, typically in SensorML format) and getObservation (used to recall data, typically returned as SensorML) along with a number of non-mandatory functions. The SOS uses the concept that an observation is an event that produces a result, the result being an estimate of the observed phenomenon. Each event is classified by a time stamp, a feature of interest, the observed phenomenon and the procedure (sensor). Data is recalled from the SOS through observation offerings, these are non-overlapping groups of related observations such as all results from a given sensor.

The development carried out for the Shale Hill Critical Zone was based on a system developed to capture environmental data within a laboratory[8]. The laboratory system uses a message broker to capture data from a number of acquisition cards and store it in an implementation of the SOS. The system was designed such that it could be scaled to work with any source and, as such, required only minor modifications to work with the varied and larger volumes of data.

2 Implementation

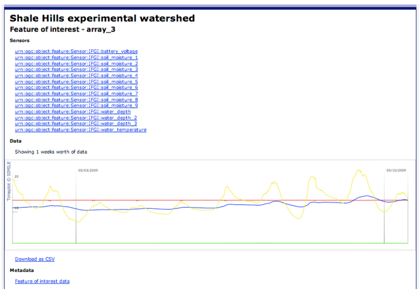
The web services developed was based on the 52°North[6] implementation of the OGC sensor observation service (SOS); this implementation is programmed in Java and runs on a Tomcat server. After a series of compatibility and scalability tests it was determined this implementation would not work with the volume and type of data from the critical zone laboratory as it was not stable to receiving data from a large volume of data producers. The 52°North implementation used a PostGIS[3] extension of the PostGreSQL[4] database to store the observed data; this database model was retained. The critical zone laboratory implementation of the SOS is split into two parts, the RESTful front end for human users to interact with and an application programming interface (API) for machine to machine interactions, both developed in PHP.

The website front end uses a structured approach to organising the data. The user is first presented with the available features of interest, these are both displayed as a list of links and, as the GML position is stored, on Google Maps although this could be used any any map system. Within a selected feature of interest the user can view the available metadata in both tubular format and as XML from the API. The Simile Timeplot[5] is used to generate a plot of all the data from the feature of interest over the previous week, although this granularity can be selected by the user. Each sensor within the feature of interest is also presented as a link. An example of a feature of interest is shown in Fig. 2(a). When a sensor is selected the metadata (as SensorML) can be viewed as well as a Simile Timeplot of previous data as with the feature of interest. Data can also be recalled from the front end, the user selects which feature of interest, sensor, start time and end time they wish to recall and a request is made to the API. This is presented to the user as a Simile Timplot and can be downloaded as CSV. An example of a sensor within the SOS is shown in Fig. 2(b). The system has been developed so each object (sensor, feature of interest, etc...) has a unique URL and therefore can be queried and addressed through simple HTTP requests.

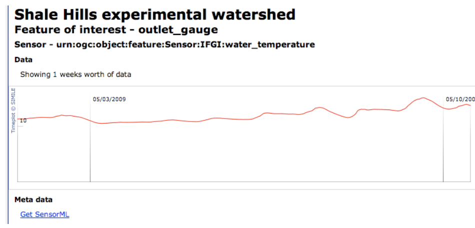
The machine to machine API simulates the operations the 52°North implementation. The API takes a single query parameter as XML and executes the request. The response is then presented to the user as defined in the SOS standards, typically as either SensorML or using Observation and Measurement encoding.

3 Discussion and Conclusions

The OGC sensor observation service provides a global standard for storing and recalling sensor data. The standard includes the storage of large amount of metadata about both the sensors (such as manufacture, model number, calibration methods, etc..) and the features of interest (such as position, bounding box, type, etc...). Allowing users access to both the observation data and metadata can increase the trust in the data that is available. The use of an API can enhance and promote data sharing between collaborators through machine to machine interactions as both queries and results will be available in



(a) A feature of interest within the SOS



(b) A sensor within the SOS

Figure 2: The website front end to the SOS

a pre-defined and known format. A web front end was generate to allow human to machine interaction through a user friendly environment.

The system has been used to capture data from the Shale Hill Experimental Watershed for a number of months. The watershed is a small area within the Shale Hills critical zone laboratory currently used to study the hydrology of the area. During this time the system captured data from approximately fifty sensors at rates varying between once every ten minutes to once every twelve hours. The data recalled from the system (through both the API and webpage) has been used to run several hydrology models of the area.

It is planned to expand the system to capture all data currently gathered in the Shale Hills critical zone laboratory, this will include a wide range of sensor type, acquisition rates and data capture methods. An overall goal of the project is to run a version of the system at each critical zone laboratory and develop a web portal to access data from all the zones through a single webpage. This will provide access to larger volumes of data than has been available for the environmental models and hopefully allow for more accurate models and results.

Acknowledgements

I would like to thank Karl Mueller, Chris Duffy, William Brouwer and Brian Bills at Pennsylvania State University for their help and co-operation. Without their help the Shale Hills critical zone data would have not been available for this research and the project would not have been possible.

References

[1] S. P. Anderson, R. C. Bales, and C. J. Duffy. Critical zone observatories: Building a network to advance interdisciplinary study of earth surface processes, February 2008.

[2] Introduction to SensorML. SensorML. [web page] http://vast.uah.edu/SensorML/. [Accessed 16 Febuary 2008].

[3] PostGIS spatial database extension for PostgreSQL. PostGIS. [web page] http://postgis.refractions.net/. [Accessed 24 July 2008].

[4] PostgreSQL: The world's most advanced open source database. PostgreSQL.org. [web page] http://www.postgresql.org/. [Accessed 24 July 2008].

[5] Simile — Timeplot. Simile. [web page] http://simile.mit.edu/timeplot/. [Accessed 12 January 2009].

[6] Welcome to 52North. 52North.org. [web page] http://52north.org/. [Accessed 12 January 2009].

[7] Welcome to the OGC website. OGC. [web page] http://www.opengeospatial.org/. [Accessed 12 January July 2008].

[8] Stephen Wilson and Jeremy Frey. The smartlab: Experimental and environmental control and monitoring of the chemistry laboratory. *Collaborative Technologies and Systems, International Symposium on*, 0:85–90, 2009.