The design of a novel time series client for ocean data.

S. J. Bainbridge
Australian Institute of Marine Science
PMB 3 MC, Townsville Qld 4810 Australia.

Abstract – **A novel time series client is described along with the supporting infrastructure required. The need for such a client comes from the lack of uptake of time series data generated by sensor network projects such as the Great Barrier Reef Ocean Observing System project. Central to the client is the need for a Registry of available data sets that can be used by the client to discover what data is available. The Registry also links the metadata and SensorML records to the data itself along with a series of fields that can be used to search for data. The client enforces a workflow that includes data discovery, data display, data downloading and export and data exploration. The development of a client that meets these needs will rapidly increase the uptake and use of environmental sensor data.**

I. INTRODUCTION

The proliferation of observing systems and technologies means that there is now more data being collected about the marine environment than before but much of this is still hard to discover and hard to use. A number of marine 'portals' are under development [1] but most of these are map based and so best deal with data that forms layers much like a GIS rather than time series data which may only be at one spatial location. At the same time standards for making time series data available, such as the Open-GIS Sensor Web Enablement (SWE) [2] mean that it is now possible to 'expose' time series data easily in a standards based manner. This paper describes the functionality of a novel theoretical time series client that looks to make the discovery and use of this growing data easier and so allow more people to use the collected data.

II. REAL TIME SENSOR NETWORK DATA

As part of the Great Barrier Reef Ocean Observing System (GBROOS) project under the Australian Integrated Marine Observing System (IMOS) project [3] a series of sensor networks are being deployed at seven sites along the Great Barrier Reef (GBR) in north east Australia. The sensor networks return large amounts of real time data of above water and in water physical parameters. This data has numerous uses from scientists wishing to understand the processes that drive reef systems to verification of remote sensing data and as input into modeling systems.

Although the data has great use the uptake and use of the data has been slow. This is primarily due to the fact that it is hard to find the data and then to undertake basic display and analysis to both see if the data has use for the issue at question and to find the patterns and information within the data.

The GBROOS project is not unique, there is a rapid uptake of sensor network technologies with a resulting increase in the amount of real time data now becoming available.

The development of sensor network technologies has in turn promoted the development of a set of standards and protocols for sensor data by the Open GIS Consortium (OGC) called the Sensor Web Enablement (SWE) 'stack' [2]. The SWE set of protocols includes standards for the description of sensors and sensor systems, standards for data transmission and standards for alerts from sensors. Some of these standards are starting to mature with a number of tools and implementations available but others are still under development.

What this does mean is that there are now standards for the basic components of real time sensor data such as data transmission (via Observation and Measurement Markup Language [4]) and sensor description (via Sensor Markup Language – SensorML [5]) and some implementations of these such as the Sensor Observation Service (SOS) via software projects such as 52North [6] and the OOSTethys Project [7].

These standards work to make it possible to 'expose' real time sensor data as standards compliant web services, what is missing is a client to deal with the data and some supporting components to aid in the discovery and use of the data.

III. FUNCTIONAL REQUIREMENTS

The client described here requires more than just a source of data, it must provide a complete end to end set of functionality for the user to find and use time series data.

The required functionality includes:

- Data discovery;
- Data display;
- Data downloading and formatting;
- Data exploration.

The first step is to be able to discover data. Most data users will not know what data is available, where it is, how to connect to it and so on. The client must be able to do a simple query and return the available data sets based on the query. The user must be able to then access the ancillary information that describes the data set such as the metadata, SensorML descriptions, data limitations and so on. The idea is to provide the user with enough information to find the required data and then to see if the data is of use.

The next step is to be able to display the data. This allows the user to first see if the data is of use and secondly to quickly see the basic information within the data.

The third step is data downloading and formatting. This maybe just simply a file download or maybe a more complex gridding of differing data against a common time base.

978-1-4244-5222-4/10/$26.00 ©2010 CROWN

The final step is a somewhat amorphous data exploration functionality that may include 4D displays, integration into modeling and visualization systems and so on.

Each of these steps needs to be completed within the client using a workflow that drives users from data discovery to basic display to downloading and saving the data. This embedded workflow makes it easy to use the client much in the way that 'wizards' drive users through complex operations by asking a series of simple questions.

IV. Components of the Data Infrastructure

Having the data available as a standards based web service is just one part of the overall system that is required. The metadata record and SensorML record also need to be available as web services so that they can be linked into the data.

The 'glue' that binds all of these services together is a registry. The registry holds details of the data set against which the client search can be run, it also holds links to the metadata and SensorML services, the actual data service that will deliver the data as well as other details about ownership, access rights and so on.

Another part of the infrastructure are external 'helper' services that are able to enhance the usability of the system. These may include ontology or semantic services that allow users to search in words that are familiar to them, language and units conversion services that make the system multi-regional and statistical and analytical services that may process the discovered data into new products or formats.

The basic infrastructure is shown in Fig One, some services may be called via the registry, others directly by the client as required.

V. Registries

From the system diagram the Registry is the main component that links the system together. There is a growing realization that registries are useful in promoting data discovery but no clear consensus as to what information they should contain and how they should be structured.

One approach is to use the metadata record and build a search engine around this. The problem here is that there are many metadata standards and even within a standard many profiles or implementations of the standard. Typically the metadata standard does not contain links to SensorML records and may be lacking other data used for the search. The most serious problem is with the granularity of the metadata record which may not match the searching requirements. Most searches for data will need to be at the sensor level, metadata at higher levels, such as the project level, is of little use if there are no lower level records describing the data itself.

Another approach is to use the getCapabilities of the Sensor Observation Service (SOS) server and this is the approach taken by the registry used by the OOSTethys project [7].

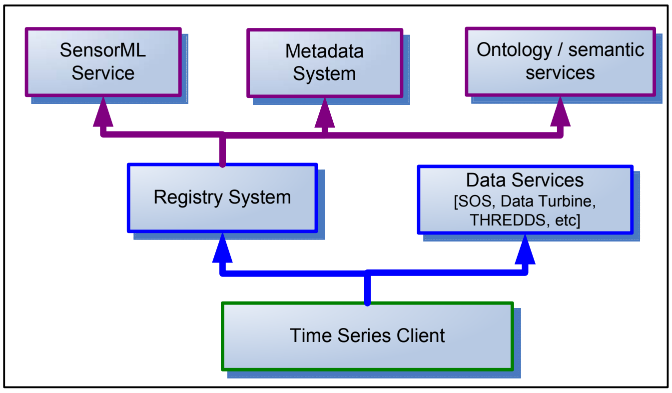

Figure One. Infrastructure required to support the time series client.

This is relatively easy in that once the registry knows the base address of the server it can harvest the details of the data including links to a SensorML record describing the sensors. The issue here is that there is a lot of required data that is not in the getCapabilities which is primarily designed to describe how to access the data and does not necessarily include information about access rights, ownership and so on.

What is required is some middle ground that combines the detailed information held in the metadata records with the detailed sensor information given by the getCapabilities function of the SOS server.

The registry needs to be simple and to support the searching capabilities for the client, it needs to include linkages to the data service, metadata service and SensorML service as well as any other services. It needs to implement access and ownership controls and requires mechanisms for the easy registration and updating of content.

The basic content would include:

- Designated fields for the main searching functionality (such as the parameter and units of measure, start and end dates, location of data service (lat/long) and depth of the data);
- URL/URI links to the data service including the format of the data service to support multiple data service types (such as SOS servers, THREDDS servers, Data Turbine, etc);
- URL / URI links to the metadata record including the standard of the metadata and a style sheet for display;
- URL/URI link to the SensorML record;
- Ownership details, who collected the data, who owns it, what access restrictions are in place.

There has been an enormous amount of work done on registries and a number of standards exist. This tends to be a complex area and one that, without some boundaries, could take years to work through. What is required is a simple workable standard that demonstrates the value and use of a registry and from there work can start on developing a full time series data registry.

VI. The Client

The client is the hub of the system in that it must provide an easy to use intuitive way not only to find time series data but ways of doing simple analysis that allow people to explore and use the data. The aim is to provide a (near) complete solution and so remove the need to use spreadsheets or other tools but to stay within the client.

The initial vision is that the client will be web based with a series of screens that drive the user through the workflow process at least for the initial steps. The first screen will be that of data discovery, that is the first part of the workflow is discovering the data. Most of this screen is made up of a simple search interface. The typical query is to find data for parameter X in region Y for time Z and maybe also for depth D for marine data.

The search interface therefore has the following options:

- A list of available parameters updated from the Registry, the user selects the one of interest;
- A list of dates for which data for the selected parameter is available again from the Registry, the user selects start and end dates as required;
- A list of depths for data in the registry that matches the selected parameter and dates, again the user can select a depth range as required;
- A map showing the location of the datasets that match the current search criteria, the user can move a bounding box to select particular areas;
- A counter showing the number of datasets in the registry that match the selected criteria, this changes as parameters are selected.

The user then searches the registry and is returned a list of datasets. The list contains links to the metadata and SensorML records allowing the user to drill down on particular datasets of interest. Data sets can be selected and added to a 'cart' for carry over to the next screen. Previous search parameters and found data sets can be saved and re-loaded for future sessions.

The second screen is a display screen and the envisaged functionality is much as per the RDV software [8] with the idea that the user can do basic data exploration within the tool. Datasets can be graphed and the display can be 'zoomed' in (both domain and range axis) and 'played' forward and backwards in time. Multi-graphs can be created by dropping data streams onto existing graphs with the option of plotting them on the same or separate axes.

It will be possible to time shift data streams so that data from differing time zones or even years could be overlaid, it will be possible to create virtual data channels from existing ones (such as subtracting one data stream from another) and to do simple statistics such as plotting trend lines and calculating co-variance between time series.

The emphasis on the display screen is to explore the time component of the data and to be able to overlay differing data to see the relationship between data sets.

The third screen is a download screen. While the intent is for the client to be able to deliver most of what the user requires there will always be a requirement to export or download the data. The download screen will allow the user to pick the data sets to be downloaded, apply any transformations to them before download (such as time shifting or creating virtual channels) and then select if the various data streams are to be moved onto a common time grid. This latter functionality is important (if computationally complex) in that the users typically require the data registered to the same time grid (say every 30 minutes). This is one area that may best be done by an external service as it is computationally complex to merge large datasets to a common time base.

The final screen is data exploration. The intention is to have complex visualization and analysis tools such as 4D playback of multiple data streams much as can be done in Google Earth [9] or other 4D tools.

VII. Conclusion

There is a fundamental gap in the end to end story of marine data management and use. There has been a justified focus on bottom up in building robust valuable data sets, in developing good metadata and recently in exposing this data as web services and SensorML descriptions. What has been missing is a sophisticated client that allows for the discovery, display, downloading and exploration of marine sensor data. This paper describes the framework that would be required to build such a client and goes through a detailed description of how such a client would work and what supporting infrastructure would be required. The aim is to make it easier for people to find, explore and download marine environmental data.

Acknowledgment

The GBROOS Project is part of the Australian Integrated Marine Observing System (IMOS) project and is funded by the Queensland State Government and the Australian Federal Government through the National Collaborative Research Infrastructure Strategy (NCRIS).

References

[1] IMOS 2010. *IMOS Ocean Portal* http://imos.aodn.org.au/webportal/
[2] OGC, 2010. *Sensor Web Enablement Working Group.* http://www.opengeospatial.org/projects/groups/sensorweb
[3] IMOS 2010. *Australia's Integrated Marine Observing System* http://www.imos.org.au
[4] OGC, 2010. *Observations and Measurements.* http://www.opengeospatial.org/standards/om
[5] OGC, 2010. *OpenGIS Sensor Model Language (SensorML)* http://www.opengeospatial.org/standards/sensorml
[6] 52North 2010. *52North Sensor Web Community* http://52north.org/maven/project-sites/swe/
[7] OOSTethys 2010 http://www.oostethys.org/
[8] NEES 2010. *RDV – Real Time Data Viewer* https://www.nees.org/it/software/rdv/
[9] Google 2010. *Google Earth* http://earth.google.com/