Oceanographic Data Provenance Tracking with the Shore Side Data System

Michael McCann and Kevin Gomes

Monterey Bay Aquarium Research Institute,
7700 Sandholdt Rd, Moss Landing, CA, USA
{mccann,kgomes}@mbari.org
http://www.mbari.org/ssds

Abstract. The importance of tracking the provenance of electronic data becomes apparent when data set providers need to also provide metadata describing where the data came from. This need has driven the development of a practical oceanographic data provenance system at the Monterey Bay Aquarium Research Institute. MBARI's Shore Side Data System is designed to manage data collected, processed, and archived from oceanographic observatories. We describe the provenance tracking aspects of this system and the lessons learned from its implementation in an operational environment.

Keywords: Ocean Observatory, Data Processing, Data Provenance.

1 Introduction

Ocean scientists collect measurements of environmental parameters with a variety of instruments and platforms. Data produced by deployments of these assets are archived and later processed for scientific analysis. The collection, analysis, and archive of data is traditionally unique for each particular kind of asset. These *stovepipe* data systems make interoperability difficult and impede the goal of gaining a more complete and timely understanding of oceanographic processes. Operating these observational assets within the context of a managed observatory is one approach the oceanographic community is using to address this problem [1]. Having the data professionally managed within regional scale observatories provides opportunities for transformative uses as described in [2]. As part of the growing effort to establish oceanographic observatories the need for improving this situation is being recognized and systems have been developed employing various methods of provenance management.

Much of the research and development in provenance systems centers on the provenance of workflow systems in Service Oriented Architecture (SOA) and grid computing environments. Friere *et al.* describe the VisTrails system [3] where provenance tracking is integral to a data workflow/exploration/visualization system. It has been used to explore simulations of coastal oceanographic processes for the Columbia River system. Observational data are used as input for the simulations and provenance is tracked from there forward. The utility of VisTrails

J. Freire, D. Koop, and L. Moreau (Eds.): IPAW 2008, LNCS 5272, pp. 309–322, 2008.

is that workflow paths may be queried for and executed again on different input data and that it may be used in many domains. Chapman and Jagadish [4] propose eleven desiderata for provenance systems and relate how several real-world systems do in terms of meeting these desiderata. Comparison of these systems with the one described in this paper is provided in a discussion section below.

The Monterey Bay Aquarium Research Institute operates several oceanographic observation platforms for purposes of advancing knowledge in the deep waters of the worlds oceans. Data collected from these systems and the experience gained from their management led to the design and development of the Shore Side Data System (SSDS). One of the requirements stated for SSDS is to *capture and archive processed data products and associated metadata, maintaining known relationships between data sets*. This capability was designed and implemented for SSDS without awareness of the wealth of other work in the field of provenance. Putting this system to work for real world problems provides an experience case for device-to-dataset oceanographic observatory data provenance. In conjunction with MBARI's mooring system, the capability of SSDS to capture the provenance metadata from device to data set makes it unique among these other systems.

In this paper we describe the Monterey Ocean Observing System (MOOS) of which SSDS is a component, the SSDS data model, its application framework, and the operational procedures required for collecting *good* provenance information. We finish with comparisons and contrasts to other provenance systems.

1.1 Monterey Ocean Observing System

In 1999, MBARI began developing the components of MOOS [5], an advanced, integrated ocean observatory. Development focused on several themes:

- set common standards for power and data management,
- allow seamless reuse of instrumentation packages across a variety of platforms,
- permit very flexible geographic instrument location, and
- do all this at an affordable cost for both system acquisition and maintenance.

MOOS development now incorporates several major components, including observatory-scale science experiments, autonomous underwater vehicles (AUVs), standalone moorings with connected benthic components, an operational at sea software infrastructure (SIAM) [6], and a data management system (SSDS).

Fig. 1 is an artist's rendering of an actual observational campaign that took place in Monterey Bay in the summer of 2000 [7]. A variety of sensors were deployed on a diverse set of platforms including aircraft, ships, remotely operated vehicles, towed vehicles, moored and drifting buoys, and autonomous powered and gliding vehicles.

Immediately following this campaign most of the original data produced from the platform deployments were collated into a directory structure and descriptions were written using the Federal Geographic Data Committee (FGDC) Content Standards for Digital Geospatial Metadata [8]. The data and metadata

Fig. 1. Artist's rendering of intensive observational campaign to observe the waters of Monterey Bay with a variety of platforms. Data from these platforms are transferred to shore for processing and in some cases products are transferred back to shipboard teams.

descriptions are available on the MOOS Upper-water-column Science Experiment (MUSE)web site [7]. For most of us who have dealt with the management of oceanographic data this exercise was our first exposure to FGDC. The experience was useful as it helped guide the design and development of the data system that would satisfy the MOOS data handling requirements. For MUSE we had the personnel resources immediately following the field program to manually assemble all the deployment and processing information into FGDC descriptions for each data set. For future MOOS operations we wished to accomplish this task more efficiently and to do this we needed a central catalog for tracking deployment and data processing information.

1.2 Shore Side Data System

Following the definition of the MOOS goals, a workshop was held to help define the requirements for its Shore Side Data System [9] [10]. The high-level functional requirements of SSDS are to:

- Capture and store data from MOOS data sources
  (Includes files and streams)
- Capture information (i.e. metadata) about the stored data
  (Location, instrument, platform, data format, etc.)
- Capture and store data products
  (Derived products, quality controlled data, plots, etc.)

- Provide access to the original raw data
- Convert data to common formats for user application tools *(Excel , Matlab, Ferret, ArcGIS, etc.)*
- Present simple plots of any well-described data
- Capture and archive processed data products and associated metadata, maintaining known relationships between data sets
- Provide access to data and metadata through an application program interface (API) and a web interface

The SSDS development team consisted on average of two domain experts skilled in software development and two software engineers experienced in developing complex systems. We followed a modified Agile software development process [11] using Java, Enterprise Java Beans, Hibernate, and JBoss for the core components of the system. The system consists of two relational databases: one to store the original instrument data and another to store all the information about the data (the metadata). It is a mixed data system with much of the processed data existing outside the database in NetCDF [12] files stored on disk but accessible through the OPeNDAP [13] web interface. References to these data products and other resources are stored in the metadata database. Fig. 2 shows the overall architecture of SSDS. Data and XML descriptions of the data flow in from the Wet Side into a high-availability Ingest component. Data and metadata are accessible via web based services.

After about two years of development the SSDS had its first deployment with some components of the MOOS SIAM software with the Center for Integrated Marine Technologies [14] mooring in Monterey Bay. SSDS has continued to operate since then and we have adapted data streams from two other operational moorings and data file processing from our AUVs. [15]

1.3 Operational Details

The beginning of any observational data path is the definition of the instrument that produces the data. With SSDS this information is stored the Device table of SSDS's Metadata database. We use XML for the exchange of the device attributes. Fig. 3 shows one of the configuration panels from one our our tools for defining the device metadata.

For Programmable Underwater Connector with Knowledge [16] equipped instrumentation this XML can be embedded in the instrument. Upon first deployment on the system data and XML descriptions of the instrument's data flow into SSDS and are made available for shore side processing. Data and metadata are accessible via web based services. Our experience of writing the FGDC metadata for the MUSE campaign taught us that the human effort of writing fully described metadata would be huge for an observatory composed of hundreds of instruments producing data. Our goal with SSDS is to provide mechanisms to automatically capture data processing metadata such that all of our data sets are described with appropriate metadata.

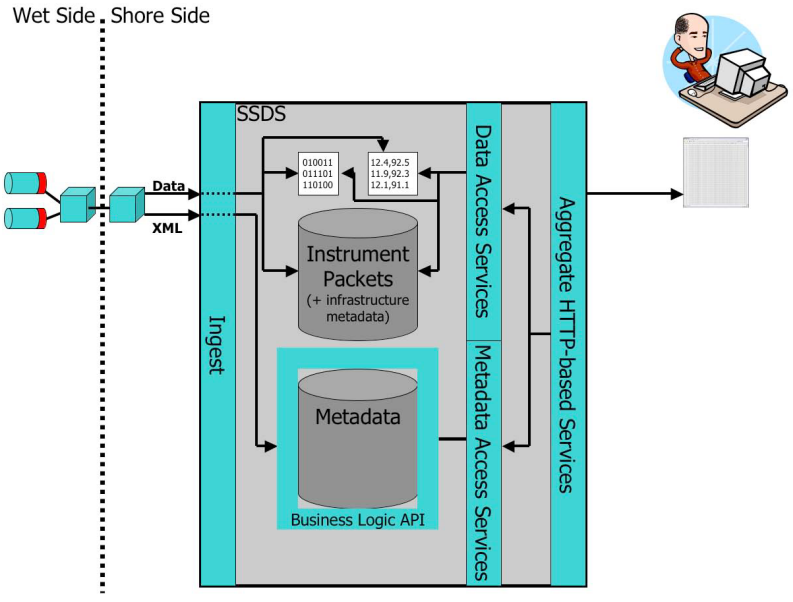

Fig. 2. The main components of the Shore Side Data System. Applications consuming data and producing derived data products read and write provenance metadata through the Aggregate HTTP-based Services and the Metadata Access Services.

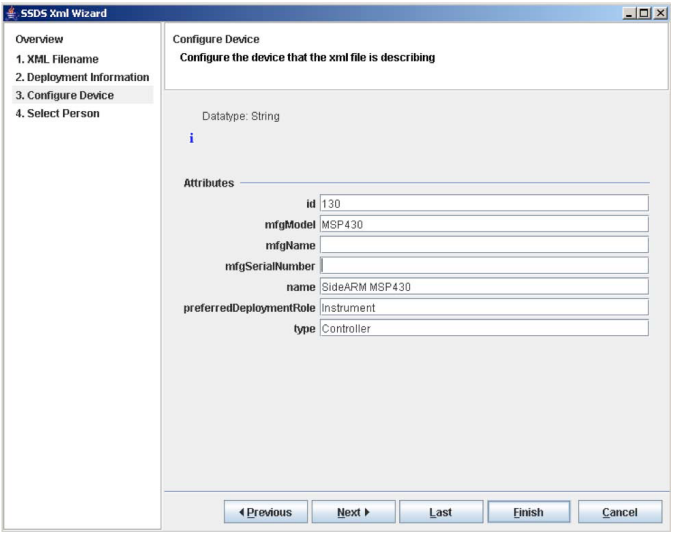

Fig. 3. Device metadata authoring tool. Details of the instrument and of its deployment are captured in advance of the deployment. Processing programs are then able to use this information to generate well described data sets.

2 Data Model

An early step in the design of SSDS was the creation of a data model that defines the entities and relationships within an operational oceanographic observatory. The diagram in Fig. 4 is SSDS's class diagram which closely parallels the table structure in SSDS's Metadata database. The main classes that are of interest for data provenance are the DataProducer and DataContainer classes.

For DataProducers that are Deployments parent-child relationships can be defined such that a sensor can be deployed on an instrument, and that instrument (along with other instruments) can be deployed on a platform. Defining these relationships is best done pre-deployment before any data are produced by the deployed system. This simple model has proven very durable. It is a general model for which any of the metadata about the data produced from the platforms shown in Fig. 1 may be stored.

Deployment and DataContainer records for the instrument deployments track important details that are needed by data processing and visualization software. Attributes of these classes are show in Fig. 5. Other important details (not shown here) are in the RecordDescription and RecordVariable classes. These classes contain parsing information, variable names, standard variable names,

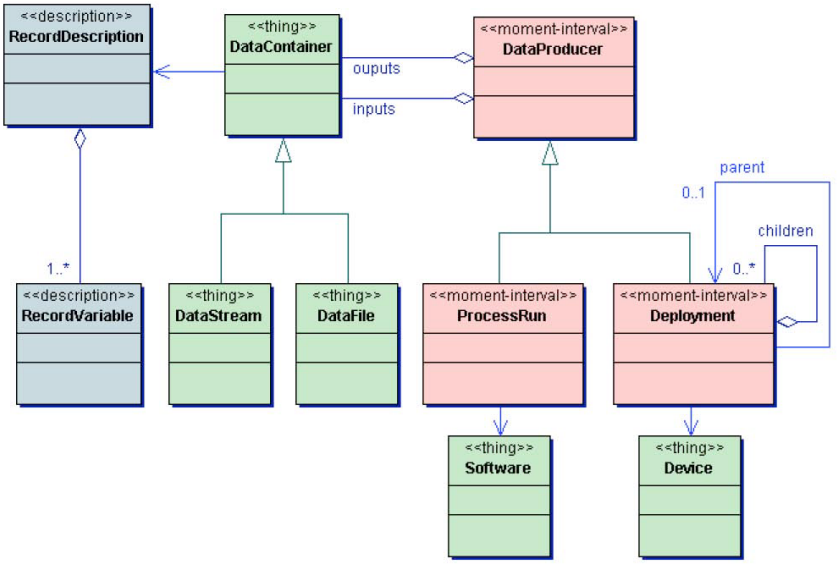

Fig. 4. Object model for the core objects in the Shore Side Data System. A DataProducer can be either a Software ProcessRun or a Device Deployment. Device Deployments may have parent-child relationships. DataProducers may output DataContainers. ProcessRuns have DataContainer(s) as input(s) and DataContainer(s) as output(s). A DataContainer is related to a single DataProducer and is described with a RecordDescription and a collection of RecordVariables. Each object has many attributes that define relevant who, what, where, when, and how information.

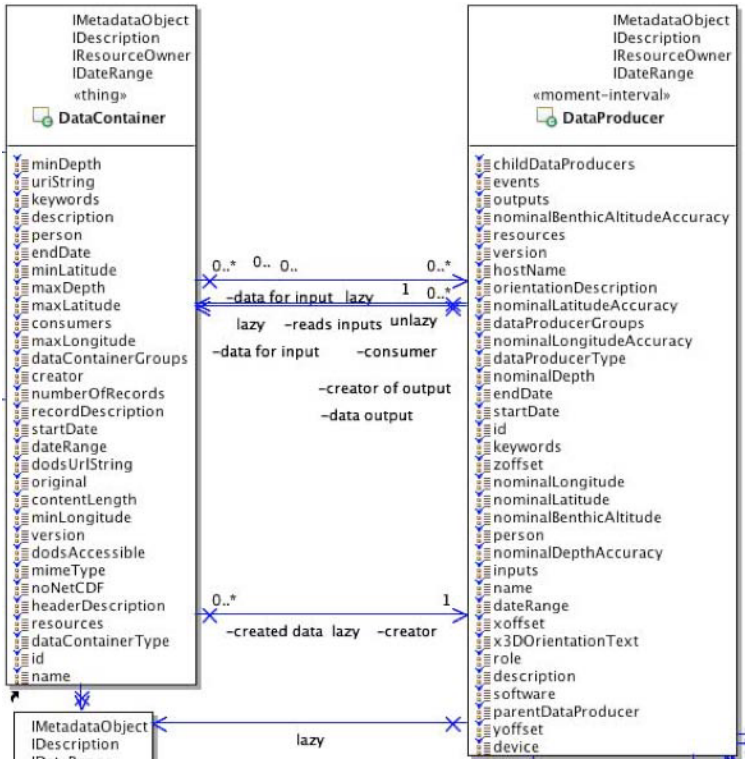

Fig. 5. Attribute details for the DataProducer and DataContainer classes

and units. Automated processing software has been abstracted such that one code base may be used to process output from any instrument. Our pre-SSDS data processing software was embedded with specific deployment and instrument deployment metadata making it cumbersome to maintain. Our current SSDS-enhanced data processing code base is much more flexible and easier to maintain as it retrieves all the information it needs from SSDS's Metadata database.

DataProducers that are ProcessRuns may have input DataContainers and their output DataContainers may be consumed by other ProcessRuns. Data from an observatory may go through several steps of processing before being provided to a scientist for analysis. SSDS can track all of these steps. There is no limit to the number of links in the chain of DataProducer-DataContainer.

3 Application Framework

To be successful as a data system SSDS had to meet the functional requirement of *providing access to data and metadata through an application program interface (API) and a web interface*. There are at least three different APIs to

SSDS Metadata: the Java Data Access Objects (optionally wrapped with Enterprise Java Beans for remote access), SOAP Web Services, and a simplified Representational State Transfer (REST)-style web interface.

3.1 Java Objects

In the Business Logic API surrounding the Metadata database a programmer can write Java code to use the Data Access Objects for interacting with the database directly through Hibernate. This approach is more attractive than writing SQL to interact with the database directly as business logic is incorporated in the DAOs making it easier to deal with the constraints and relationships within the database. It also abstracts out the underlying database technology so the database provider can be changed (MSSQL to MySQL, for example). This API is used for common core functions operating within the J2EE environment on the SSDS server.

3.2 Perl Module

Much of the routine data processing done in oceanographic research environments is done with powerful script languages such as Matlab, Python, Perl and Ruby. Of these Perl and Matlab are the most frequently used by people at MBARI. One of the products of SSDS is a Perl module (SSDS.pm) that is automatically generated during an SSDS build from reflection on the Java classes. (This technique may be used to generate native libraries for other languages as well.) This simplifies access to SSDS Metadata from Perl scripts. For instance, printing the names of the devices deployed on the 'M2 - May 2005' mooring deployment is done with this code:

```
#!/usr/bin/perl -w
#
# Print the child Deployments of mooring deployment name M2 - May 2005
#
use SSDS;
$ssds = new SSDS();
$ssds->ssdsServer('http://ssds:8080/');
my $dpAccess = new SSDS::DataProducerAccess();
my $dps = $dpAccess->findByName('M2 - May 2005', 'true', 'name', 'false');
my $children = $dpAccess->findChildDataProducers(${$dps}[0], 'name', 'false');
foreach my $child (@{$children}) {
    print "child deployment device name = " . $child->getDevice()->name(). "\n";
}
```

This little program produces output like this:

```
child deployment device name = OASIS3 Controller
child deployment device name = Garmin GPS
child deployment device name = Workhorse Long Ranger ADCP
child deployment device name = MicroCAT Serial CTD
child deployment device name = MicroCAT Serial CTD
child deployment device name = Biospherical PRR Spectroradiometer
```

```
child deployment device name = pCO2 Analyzer
child deployment device name = MBARI ISUS
child deployment device name = MBARI Metsys
child deployment device name = WETStar Fluorometer
child deployment device name = E-meter
child deployment device name = HydroScat-2
child deployment device name = Surface Inductive Modem
child deployment device name = HOBI HS2
```

Though we are showing just the name of the device deployment in the above example. all of the attributes from all of the objects are available to the script.

An example of how our operational software traverses the metadata structure to produce a fully described mooring data set follows.

Starting at a parent mooring deployment as in the one above do the following:

- Loop through all the child device deployments and find output data container references.
- Build structures of variable names and units from the RecordDescription for the DataContainer.
- Parse the records from the DataContainer and create a Climate Forecast [17] compliant NetCDF data set.
- Populate discovery level metadata by walking the deployment tree as necessary. For example, the nominal position of the mooring is an attribute of the mooring platform deployment and not of the individual instrument deployments. If the Deployment that produced the data does not have the information (e.g. nominal depth, latitude, longitude) then the code walks up the parent deployment heirarchy until it finds the needed attributes.

When the Perl script has completed generation of a new DataContainer it calls the SSDS DataProducerAccess functions to add (or update) a ProcessRun. The script has access to all the information that is needed to populate the attributes that help provide fully tracked provenance information. A key piece of information is the specific version of the code that generates the output DataContainer. This is done in coordination with our version control system and the $Id:$ expansion within the source code. The specific version is parsed from the source code and the Version attribute for the Software object associated with the ProcesRun is set to the value from the expanded $Id:$ line. Discipline is required in making sure to commit changes before running the software; otherwise the version number parsed from the file will not accurately reflect the actual version used for the processing.

3.3 Web Application

Having visibility into the relationships and attributes of all the objects in the SSDS Metadata database is essential for being able to maintain the integrity of the data. One such tool is SSDS Explorer, a Java web application that allows

a user to navigate down device deployment trees and down chains of data processing. Along the way attributes and references to resources are made available for examination. An example view of an expanded tree is in Fig. 6. The highlighted output DataContainer, a netCDF file named tenMinuteM2_20050520.nc was produced from a run of the combineAll.pl script on 28 August 2007. There are two other scripts (DStoNetCDF.pl and combineTS.pl) that operated on the data after it was produced by a MicroCAT Serial CTD instrument deployment on mooring M2.

4 Operational Procedures

SSDS requires the definition of instruments and the data they produce before the instruments are deployed to collect data. Having tools (Fig. 3) and clearly documented procedures that complement rather than replace existing procedures facilitates the capture of instrument metadata. This is the first step for tracking data lineage and is required for its processing within SSDS. As SSDS provides value with self-described data files and automated time series plots there is additional motivation for people to provide the information the system needs. The life cycle of data within SSDS is as follows:

- Instrument is defined by creating Device record.
- Instrument is configured for deployment by writing Deployment, DataContainer, RecordDescription, RecordVariable XML.
- Instrument is deployed. XML metadata is ingested by SSDS, data packets flow into the Instrument Packets database.
- Automated DataStream processing software consumes the data packets producing a NetCDF file for each instrument's data. A DataProducer record is created linking the input DataStream to the output DataFile.
- Follow-on data processing runs consume instrument NetCDF DataContainers producing combined data sets and graphical products. Metadata from SSDS is extracted as needed to fully describe data in all the NetCDF data sets.
- User uses the data with all the needed information to assess its suitability for a particular use.

5 Discussion of Provenance Systems

As oceanographic observatories become established we can expect a variety of solutions to provenance management issues. The spectrum of problems and their solutions is wide. SSDS provides a solution to the management of provenance early in the life cycle of observatory data — between instruments and their configuration to the data sets produced by their deployments. Many of the provenance systems described in the literature operate "downstream" of where SSDS operates. They operate in workflow, SOA, and high-performance grid computing environments. SSDS operates in a less sophisticated environment where highly

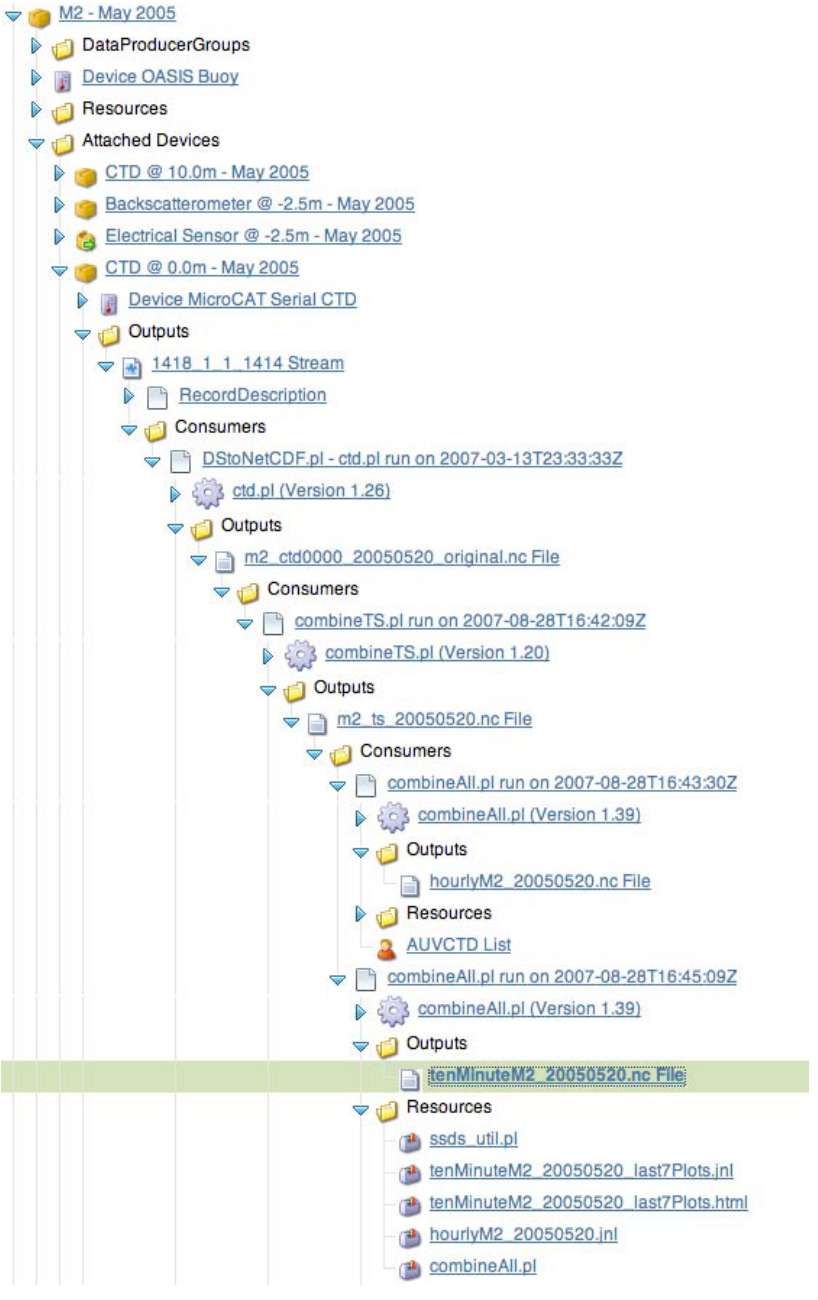

Fig. 6. Web application display of OASIS Buoy deployment's CTD Device output and the chain of ProcessRuns and Outputs that finally combine data from the other instruments into a gridded tenMinute NetCDF data set the. Ancillary output from the combineAll.pl ProcessRun are recorded as Resources.

diverse, but small volume data streams are transformed with relatively simple processing programs that are not computationally demanding.

In this environment we employ a *pay-as-you-go provenance* [2] system: as each process is executed a p-assertion [18] is constructed by provenance-aware software and is sent to the provenance store. As data sets are produced following this discipline, application software access the provenance store to generate metadata that assists in evaluating the usefulness of downstream data sets. This is a powerful capability, but it is not all that can be accomplished with a provenance management system.

Of the several provenance management systems reviewed [19], Chimera [20] and the Earth System Science Workbench (ESSW) [21] have the most similarity to provenance management within SSDS. Key to SSDS provenance are the *Software*, *DataProducer*, and *DataContainer* objects. Chimera has exactly the same concepts, but calls them *transformation*, *derivation*, and *data object*. Chimera, like SSDS, stores relationships between these objects in a relational databse. It specifies a Virtual Data Language (VDL) where details of all processing (derivations), including command line arguments, may be specified such that output data objects may be created by executing the derivation again. VDL also allows the specification of queries to search for derivations or the producers of specific data objects. Stored derivations may be re-executed on different input data, may be used to dynamically regenerate missing output files, and may be scheduled to run massive computations in distributed grid computing environments. ESSW, like SSDS, puts the onus on the script writer to construct and submit the p-assertion.

Chimera has been used to conduct analyses of data from millions of input files, producing another million output data files, from the Sloan Digital Sky Survey. Chimera handles all the file bookkeeping easing the headache that can be involved with data management in domains such as astronomy and high energy physics. A system such as Chimera or ESSW may be helpful in oceanography. Data from ocean observatories are fed into numerical simulations of ocean processes. These simulations are quite computationally demanding and do make use of Service Oriented Architectures and grid computing environments [22]. Driving Ocean Observing System Simulation Experiments from a robust data provenance management system is an attractive proposition.

6 Conclusion

The Shore Side Data System has been in operation for about four years and is meeting its functional requirements. A key feature of the system is its ability to track the specific processing steps of data processing pipelines. This capability provides detailed provenance information for all the data sets produced with SSDS enhanced software. This helps us create better described data sets with more efficient and simpler to maintain data processing software. The system is currently undergoing some refinements to improve its ability to maintain the metadata store and we are also simplifying the configuration steps so that it is

easier for others to install and use. We are not done capitalizing on the data provenance tracking capabilities within SSDS. Future projects may fund the incorporation of workflow tools and better client processing integration. We want the barrier to be low for people and software to use the provenance aspects of the system.

One of the lessons learned from our efforts is that discipline is required at key steps of the data generation and processing processes. Accurate recording of instrument attributes is required in advance of the deployment of the instrument and once it is deployed the deployment information (time, location, and parent deployment device) is required at the beginning of the data processing path. Original data is processed by software whose source code is maintained in a version control system. As the data are processed the specific version number of the software components are logged. With this system we have reproducibility of derived data from the original data and exposure of specific processing techniques used at every step of the data path. A logical next step to consider is the integration of SSDS with an industry-standard provenance management system such that more benefits of provenance management may be realized.

Acknowledgements

This research was supported by the Monterey Bay Aquarium Research Institute through funding from the David and Lucile Packard Foundation.

References

1. Glenn, S., Schofield, O.: Observing the Oceans from the COOL Room: Our History, Experience, and Opinions. Oceanography 16(4), 37–52 (2003)
2. Baptista, A., Howe, B., Freire, J., Maier, D., Silva, C.T.: Scientific Exploration in the Era of Ocean Observatories. Computing in Science and Engineering 10(3), 53–58 (2008)
3. Freire, J., Silva, C.T., Callahan, S.P., Santos, E., Scheidegger, C.E., Vo, H.T.: Managing Rapidly-Evolving Scientific Workflows. In: Moreau, L., Foster, I. (eds.) IPAW 2006. LNCS, vol. 4145, pp. 10–18. Springer, Heidelberg (2006)
4. Chapman, A., Jagadish, H.V.: Issues in Building Practical Provenance Systems. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering 30(4), 38–43 (2007)
5. MOOS: Monterey Ocean Observing System, `http://www.mbari.org/moos/`
6. SIAM: Software infrastructure and application for MOOS, `http://www.mbari.org/moos/siam/siam.htm`
7. MUSE: MOOS Upper-Water-Column Science Experiment, `http://www.mbari.org/muse/`
8. FGDC: Federal Geographic Data Committee, `http://www.fgdc.gov/`
9. Graybeal, J., Gomes, K., McCann, M., Schlining, B., Schramm, R., Wilkin, D.: MBARI's Operational, extensible data management for ocean observatories. In: The Third International Workshop on Scientific Use of Submarine Cables and Related Technologies, Tokyo, pp. 288–292 (2003)

10. The Shore Side Data System, http://www.mbari.org/ssds/
11. Agile software development, http://en.wikipedia.org/wiki/Agile_software_development
12. NetCDF: Network Common Data Form, http://www.unidata.ucar.edu/software/netcdf/
13. OPeNDAP: Open-source Project for a Network Data Access Protocol, http://www.opendap.org/
14. CIMT: Center for Integrated Marine Technologies, http://cimt.ucsc.edu/
15. Gomes, K., OReilly, T., Graybeal, J.: Issues in data management in observing systems and lessons learned. In: Proceedings of the Marine Technology Society/Institute of Electrical and Electronics Engineers Oceans Conference, Boston, Massachusetts (2006)
16. PUCK: Programmable Underwater Connector with Knowledge, http://www.mbari.org/pw/puck.htm
17. NetCDF Climate and Forecast (CF) Metadata Convention, http://cf-pcmdi.llnl.gov/
18. Moreau, L., Groth, P., Miles, S., Vazquez-Salceda, J., Ibbotson, J., Jiang, S., Munroe, S., Rana, O., Schreiber, A., Tan, V., Varga, L.: The Provenance of Electronic Data. Communications of the ACM 51(4), 52–58 (2008)
19. Simmhan, Y.L., Plale, B., Gannon, D.: A survey of Data Provenance in e-science. SIGMOD 34(3), 31–36 (2005)
20. Foster, I., Vockler, J., Wilde, M., Yong, Z.: Chimera: a virtual data system for representing, querying, and automating data derivation. In: Proceedings of 14th International Conference on Scientific and Statistical Database Management, 2002, pp. 37–46 (2002)
21. Frew, J., Bose, R.: Earth System Science Workbench: A Data Management Infrastructure for Earth Science Products. In: Proceedings of the 13th International Conference on Scientific and Statistical Database Management, Fairfax, VA, pp. 180–189 (2001)
22. Abbott, M., Sears, C.: The Always-Connected World and Its Impacts on Ocean research. Oceanography 19(1), 14–21 (2006)