Issues in Data Management in Observing Systems and Lessons Learned

Kevin J. Gomes
John Graybeal
Thomas C. O'Reilly
Monterey Bay Aquarium Research Institute
7700 Sandholdt Road
Moss Landing, CA 95039 USA

***Abstract*-With ocean observatories growing in importance and several development efforts underway, it is critical to understand the goals and issues that an ocean observing system will have to face and solve. There are many components such as network infrastructure, instrumentation, control systems, data systems and client applications that will need to interoperate seamlessly in order for the observatory to effectively meet the needs of the research community, policy makers, and the general public.**

For example, instruments serve as the critical foundation to the usefulness of the information extracted from an ocean observatory. However, configuring these instruments for deployment in an observatory can be a time consuming and error-prone task. Although manageable on an instrument-by-instrument basis, configuration becomes a very important issue as the size of the observatory and the number of instruments it operates grows. Each instrument type also has its unique power, communication and bandwidth requirements that further complicate their integration into observatory systems. As the complexity of integrating the instruments into the observatory increases, so does the overall operating cost of the observatory, thus affecting the overall capability of the system. For this reason, adding and removing instruments needs to be as simple as possible, which necessitates that the infrastructure handle a large portion of that integration automatically. Once these instruments are successfully deployed, the infrastructure must also be able to monitor the health and status of the various observatory assets.

Once past the configuration and management issues of instruments, the observatory still faces other issues from this collection of data from these heterogeneous instruments. Metadata and data management are particularly difficult problems to handle from a systematic perspective. Being able to capture and utilize metadata and data is certainly one requirement of a data management system, but having it operate in an automated and robust way creates even more complications. The metadata associated with data must be correctly captured by the system but also maintained correctly and linked with other related metadata throughout the system. Through this metadata, the relationship between the data and its context (source, environment, location, etc) can be captured and utilized in the analysis of the data in search of various phenomenon like events, trends, patterns, etc. Adding to these complicated tasks is the requirement that this data system must interoperate with other systems as both a server and a client.

Even with these high level goals and issues defined, there is no substitute for practical experience in affirming that the right goals and issues were identified. At the Monterey Bay Aquarium Research Institute (MBARI), we have had an active ocean observing development project (Monterey Ocean Observing System-MOOS) going for the past several years. With that experience, we have been able to identify several key issues and lessons learned that are relevant to ocean observatories both from the development and operational perspectives. This paper will describe the different goals that the MOOS system and its associated data management system, the Shore-Side Data System (SSDS) address. Specifically, we will discuss how the SSDS handles various data and metadata related issues and what is gained by solving those issues. Practical examples of these solutions will be given and they will be used to illustrate how, and why, certain issues are important for the data management system to address.

As a final wrap up, a section on lessons learned will be discussed to help transition what we have learned to the general oceanographic community.

I. Introduction And Identifying Needs

To fully understand our global environment, we must be able to quickly obtain an accurate picture of the state of the ocean as well as understand its interaction with the terrestrial and atmospheric systems. This picture can only be formed by having ready access to environmental data that can give us the current state of the ocean system as well as providing us the information to attempt to predict where the system will be in the short and long term. This is critical to effectively manage the resources our oceans provide us.

As a foundation to building this picture, ocean observatories are being constructed as the means to collect data from the ocean and provide the tools necessary to effectively interpret those data. Programs such as Integrated Ocean Observing System (IOOS), Ocean Observatories Initiative (OOI), and MBARI's MOOS, are examples of efforts that are tasked to provide the infrastructure for ocean observatories.

This paper will discuss the important needs of observatories and explain why they are important and how the MBARI MOOS system addresses those needs. Lastly a section focusing on the data management aspect of ocean observatories will detail some of the lessons that were learned by implementing the MBARI MOOS and SSDS systems.

A. Configuring Instruments

The most fundamental components of the data in an ocean observatory are the instruments themselves. In order to match the large number of physical parameters in the ocean, many different types of instruments are needed to measure the various physical quantities. These instruments are created by many different manufacturers ranging from Commercial Off-The-Shelf (COTS) instruments to one of a kind instruments built in a laboratory.

No matter how varied these instruments are, they must utilize the ocean observatory infrastructure to get data into the user's hands. At some point, the instruments must be able to communicate with, and take direction from, the supplier of power and communications [1]. Without this bidirectional communication, the infrastructure will not effectively manage the instrument and supply it the correct amount of power and bandwidth. This task is complicated not only by the numerous hardware and communication protocols, but also the non-standardized instrument descriptive information (metadata) and data formats.

B. Instrument Lifecycles Are Dynamic

In any observatory, the instruments that are deployed will be in constant flux. In an operational, monitoring style of observatory, instruments will be changed out simply for normal maintenance, cleaning and also upon failure. In more research oriented observatories, instrument deployments can change depending on the science objective or experiment design [2][3].

In cases where instruments must participate in event detection and response, the instruments can also change their configuration in-situ [1]. For example, if a seismic event occurs, it may be desired that an instrument both increase its sampling rate as well as turn on other sensors that were left dormant due to limited number of samples that sensor can acquire. Thus the state and metadata associated with an instrument can change frequently. The observatory must be able to not only operate through these changes, but track them and make that information available to end users and processes.

C. System Resource Management

The infrastructure must manage system resources required by instruments [4], and give users an exact picture of resource state. As stated in [1]:

"In essence, resource management requires knowledge of the capabilities and limitations of each system resource at any time, the actual, average, and peak resource needs of each element in the system at any time, and the actual utilization of each resource at any time. And in principle, it should be possible to predict the effect of any change in the allocation of a resource on the performance of the system, including its effect on other resources."

D. Status and Health Monitoring

Closely related to resource management is the issue of infrastructure status and health monitoring. There can be several levels of monitoring that are important for different facets of the observatory. At the most fundamental level, managing power and bandwidth will keep the system in a basic operational state [1], but there are instrument specific details that can also determine the health of the system.

Each individual instrument can have custom events and parameters that can indicate the status and health of the instrument, but also provide indirect information about the connected infrastructure. These are not always considered important to the basic resource management operation of the infrastructure, but nonetheless need to be monitored to create events that can help troubleshoot problems or flag items of interest to the end user. This can be as simple as a certain variable that an instrument is measuring going out of range.

E. Metadata Must Tie to Data

It is critical that data generated by the instruments and the infrastructure itself be tied closely to the information describing that data (metadata) [1]. At a very basic level, the parties interested in the data will need the metadata to interpret the raw values coming from the source. This is often considered "use" metadata and without it, data become meaningless. The range of this data and metadata can be quite vast as stated in [5]:

"The diversity of data type that will be collected and applied to the coastal zone is large. These will include, but not be limited to, observed data, derived data and data products, and predictions and modeled data, along with appropriate meta-data."

Metadata also provide insight into data quality by identifying the data source. [4] states:

"The ability to associate metadata with data and services can be important, as can the ability to determine the identity of entities that assert metadata, so that consumers can make their own decisions concerning quality."

F. Instruments Can Cross Observatories

Some instrumentation that will be deployed in observatories can be quite costly to develop and thus be in very limited supply. To accommodate the demand for this instrument, it will need to be moved among different observatories depending on the needs of the end users [1][3]. Having the instrument be able to seamlessly plug into different observatories is critical in reducing deployment time and retooling costs. With the OOI defining three different classifications of observatories (coastal, regional, global) [2], it is clear that being able to move instruments from one observatory to another will be a common need.

G. Metadata and Data Can Cross Observatories

Being able to physically move the instrument between observatories entails more than just hardware and basic communications interoperability. The data and metadata from the instrument must also be interoperable in order to allow each observatory to effectively manage its resources and instruments. Another benefit of data and metadata interoperability is that the same set of tools [6] (e.g. observatory visualization [2]) can be used between the different observatories. As is stated in [4]:

"The key to such success is uniformity of interface, so that programs can discover and access services without the need to write custom code for each specific data source."

II. SIAM AND SSDS INTRODUCTION

As part of the MOOS program, two software components were developed to help manage the observatory infrastructure, the instruments, and the data/metadata generated by both. The observatory middleware, known as the Software Infrastructure and Applications for MOOS (SIAM), and the data management system, known as the Shore-Side Data System (SSDS), have been in development and operation for the past few years and, as of this writing, are currently managing 3 mooring surface expressions as well as one Benthic Instrument Node (BIN). The BIN is located at approximately 3200 meters depth and connected via an Electro-Optical-Mechanical (EOM) cable to a surface expression mooring. The next section focuses on the Shore-Side Data System. A parallel paper focuses on SIAM [7] and another on the NSF SENSORS workshop [8].

III. MEETING THE NEEDS

A. Configuring Instruments

The SIAM middleware is designed to provide plug and work functionality for the numerous types of instruments that plug into the MOOS observatory [7]. This plug and work functionality provides the necessary hooks to allow instruments to track and publish both their data and metadata.

Although most of the plug and work configuration occurs outside of the SSDS, an application called ACE Configurer and Editor (ACE) interfaces to the SSDS to help the users configure the metadata that describe the instrument helping to configure the instrument for deployment (see Fig 1).

After configuring the parameters with ACE, a user generates a "SIAM jar file" that contains the instrument metadata and "service" code. If the target instrument is PUCK-enabled, ACE then writes the jar file to the PUCK. Otherwise the jar file must be installed on the MOOS node that will host the instrument [7].

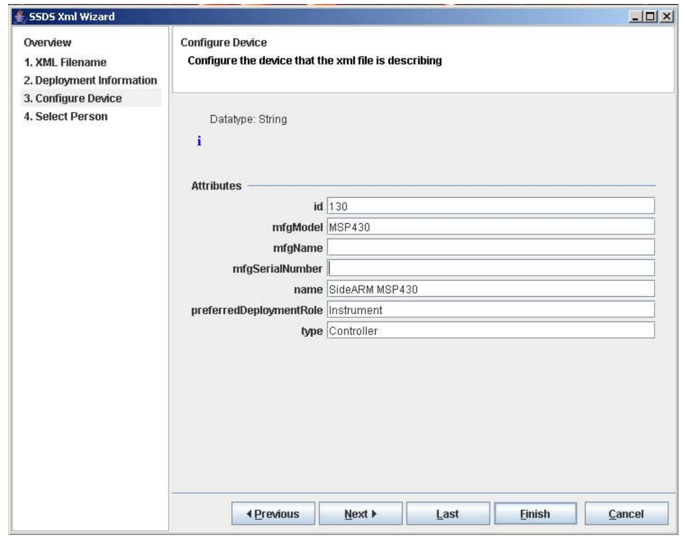

Figure 1 – ACE Application Configuring an Instrument for Deployment in MOOS

B. Instrument Lifecycles Are Dynamic

When an instrument is plugged into the MOOS observatory, its SIAM service is instantiated. The SIAM service first requests appropriate configuration and state metadata from the instrument's serial interface; the service incorporates this metadata into the instrument data stream which is later forwarded to SSDS. SSDS uses this metadata to track the different lifecycles that the various instruments go through. As this metadata moves through the SIAM system, more observatory metadata is attached so that the SSDS can extract even more information about the instruments that are located in the MOOS system. If an instrument is removed from one platform and then plugged into another part of the MOOS observatory, SSDS can track that information to give the users an idea of where an instrument has been deployed.

Fig 2 shows a query for deployments of a specific device which then shows all deployments of that instrument and the information related to that deployment such as its geospatial location. The user interface also utilizes the metadata to give the user more information about the lifecycle. In Fig 2 the user can quickly tell which of these deployments is current both by date and by the green arrow on the yellow icon next to the deployment.

C. System Resource Management

With the metadata from the MOOS observatory now in SSDS, we can get different views of the resources that are deployed. This capability gives the operations group a comprehensive view of these resources in a very short time.

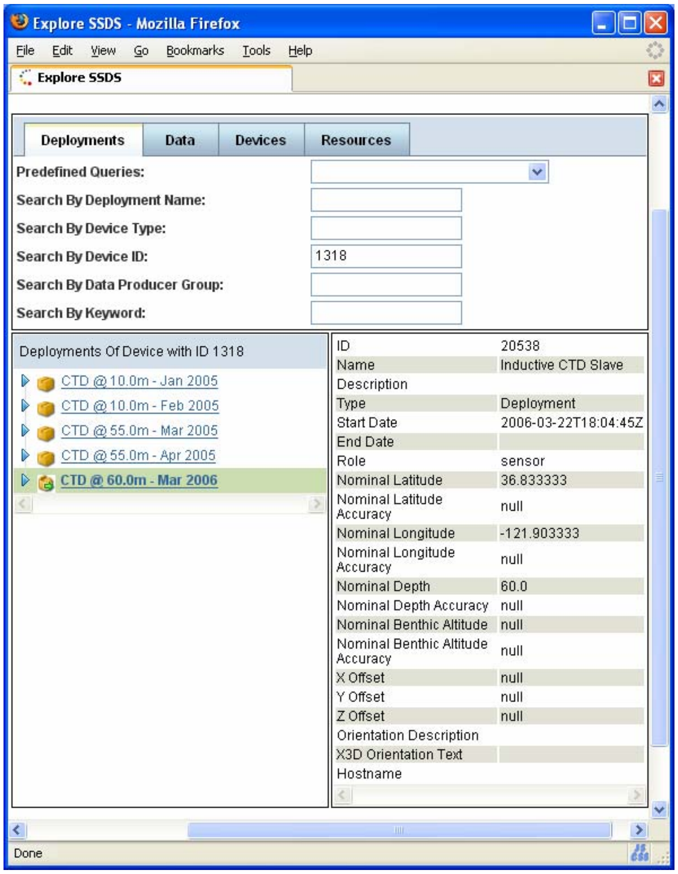

Figure 2 – The Results of a Deployment Query in SSDS

Fig 3 is a view of an SSDS query showing the resources attached to the most recent MOOS deployment, the MOOS Science Experiment (MSE). In this view, the user can see that there are four nodes that comprise the mooring and benthic deployment. In this example, the surface node is expanded to show all of the instruments that are deployed on the surface. The user can see that one of the radiometers (highlighted in the image) was replaced by a similar radiometer (further down the list). The user can tell this as the icon for the selected item shows a closed box with no green arrow (indicating a closed deployment). This view was built on the fly by the SSDS as the metadata from the SIAM infrastructure was fed into SSDS. This type of automatic view of the observatory can quickly give the user an idea of the resources that are deployed to assist in making decisions about resource management.

D. *Status and Health Monitoring*

The SIAM infrastructure provides a wealth of information to help in monitoring the resources on the observatory [7]. In addition to this, the SSDS utilizes the metadata to help monitor other parameters not covered in the SIAM infrastructure. At a very basic level, for example, the SSDS will keep track of the data streams that are reporting from instruments and notify users by email when their instrument stops reporting data. This has been used extensively by our operations group to help in the troubleshooting of both infrastructure and instrument problems.

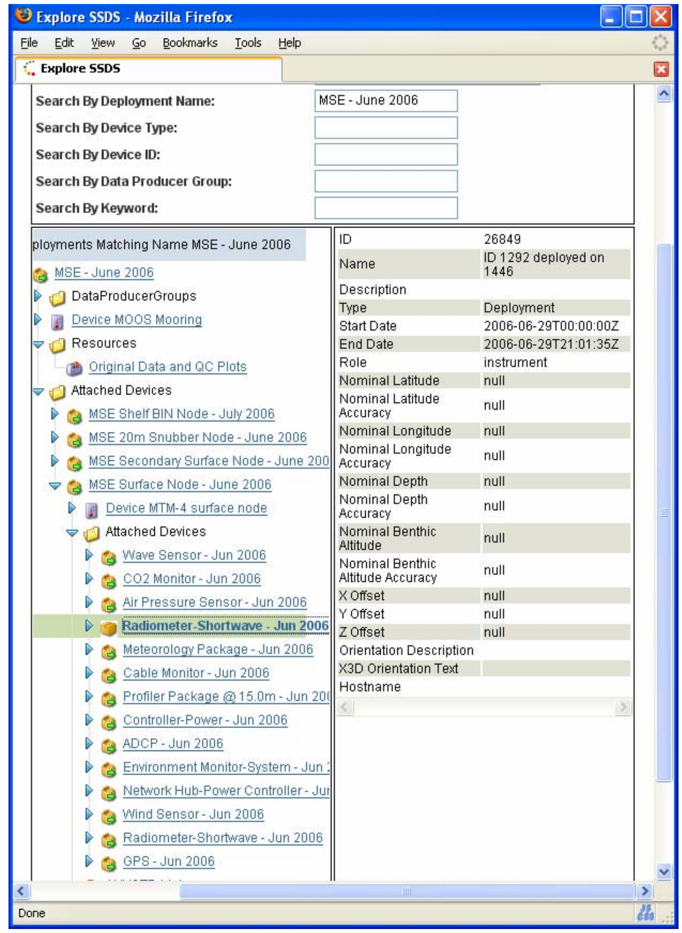

Figure 3 – Hierarchical View of the Most Recent MOOS Mooring

In addition to the basic monitoring, the metadata that is provided to the SSDS allows it to "crack" the various formats of data packets produced by the different instruments and monitor data streams to check for maximum and minimum values. If a certain variable goes out of range, the SSDS can be configured to notify a user so that data quality problems can be discovered quickly.

E. *Metadata Must Tie to Data*

One of the crucial lessons MBARI has learned over its long history of managing time series data is that tying the metadata to the data as soon as possible is paramount to the effective use of the data. At the first level of data processing, having this metadata is necessary so that the user (or process) can understand what the data contains and how it can be interpreted. Further down the processing tree however, the metadata becomes even more critical as it allows the user to ascertain the quality of the data by knowing where the data originated from.

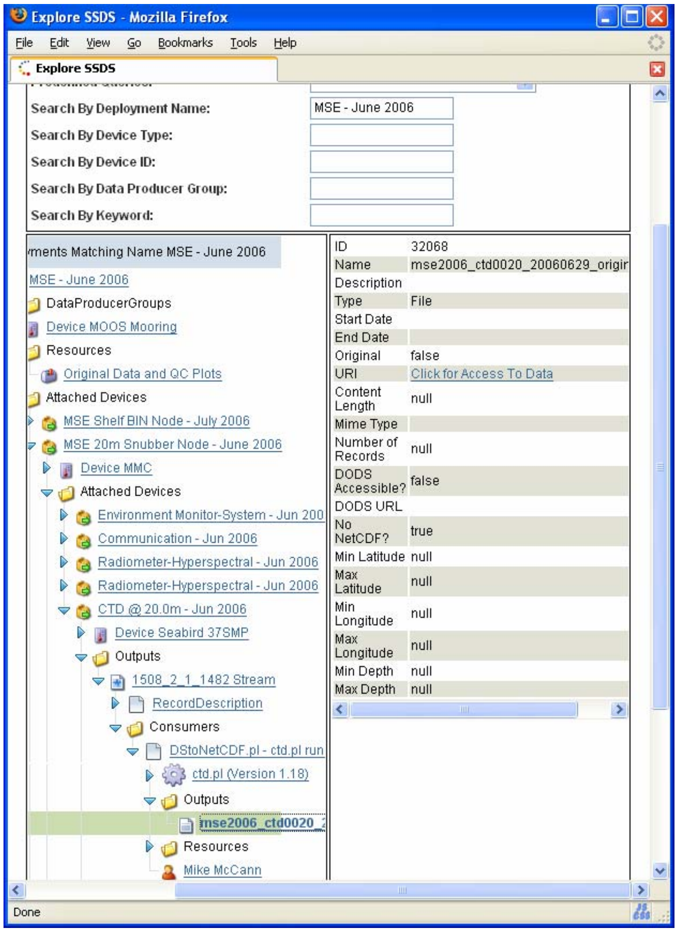

Figure 4 – A Hierarchical View of Data Post-Processing

In the SSDS, data processing histories are tracked and the appropriate metadata is tied to the data so the end user will have all the information at their fingertips to make effective use of both raw and post-processed data. If the user were to keep drilling through the nodes shown in Fig 3, they would be able to find the data and metadata that comprise the post-processing of the raw mooring data. An example of this can be seen in Fig 4. It shows a perl script that read in the CTD data from the 20m snubber node and then created a NetCDF file. The user can pull up all the relevant data and resources such as the listing of the perl code, plots created by data processing and plots of raw data, from one user interface. The reverse path is also true: if an end user finds post-processed data in the SSDS, they can trace back to see the original data, thus helping them ascertain the quality of the data product they are currently working with. This type of user interface could only be made possible by having the metadata tied to the data as it moved through each step of the process.

F. *Instruments Can Cross Observatories*

The architecture of the SSDS ensures that it is not tightly coupled with any underlying infrastructure by utilizing messaging and Service-Oriented Architecture (SOA). This means that the SSDS can be utilized to collect data from various instruments and even observatories. When an instrument moves from one observatory infrastructure to another, it can utilize the same messaging and HTTP/SOAP based protocols to communicate with SSDS, thus making the instrument movement between observatories seamless.

As a test of the portability between observatories, an instrument and its SIAM services were moved from the MOOS observatory infrastructure to the Monterey Accelerated Research System (MARS) infrastructure, which is operating in pre-deployment mode. Without any modification, the SSDS was able to consume and utilize the data and metadata from the MARS observatory in the exact same manner as was shown earlier in the MOOS observatory. This is important as it shows that if the different components of an observatory infrastructure utilize various standards to communicate, these components can be interchanged between observatories.

G. *Metadata and Data Can Cross Observatories*

As mentioned in the previous sections, if standards are used to define interfaces between different components, it helps in the reuse of the components. In addition, however, it also opens the door for integration with existing tools that may not even be a part of the domain of interest.

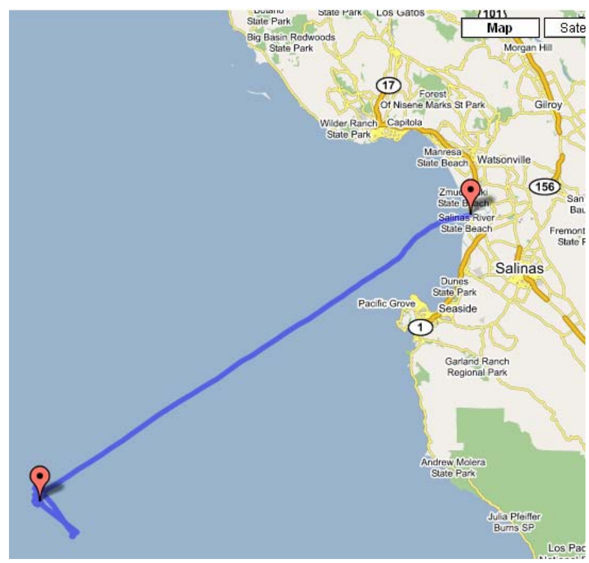

Figure 5 – GPS Track of Mooring Deployment Displayed in Google Maps.

As an example, because we use a standard schema to describe the metadata coming from instruments, we can transcribe data from custom formats into forms that are expected by other tools. In Fig 5, SSDS has taken the GPS data from the most recent mooring deployment and utilized Google Maps to plot the track of the mooring deployment.

In addition to leveraging tools, standard interfaces also allows other data providers to send their data and metadata to the SSDS. Although the AUVs that MBARI is currently operating are not running the SIAM infrastructure, by utilizing the standard interfaces, the AUVs can be folded into the observatory in the SSDS. MBARI users can now access both AUV and MOOS data through the SSDS, making all this information appear as a "virtual" observatory.

IV. Lessons Learned

During the MOOS program, there have been several cycles of deployment of observatory assets and instruments. From these iterations, we have been able to learn many things, some of which we incorporated into design changes, leaving others for future work. In this section, we briefly discuss some of those lessons and why they are important.

Although not necessarily a unique lesson, we realized quickly how important solid interface definitions were. The MOOS System is a collection of hardware and software components that were developed by different groups on different schedules. The only way to really ensure that these components would work together in the end was to have interfaces that were tightly controlled. Unfortunately, in some cases interfaces changed, which caused some unnecessary iterations or design changes. This is not to say that these components are tightly coupled, just that their interfaces and the change control process must be rigorous so that the component developers can develop against a contract.

Web services have been gaining momentum as the choice for distributed systems in commercial and cyberinfrastructure architectures [9]. We found that web services indeed work well in supporting loosely coupled architectures. In addition to this, we found that while these services helped in connecting systems, users were not always enthusiastic about working with the traditional web services stack (SOAP over HTTP). In fact, our most widely used interfaces in the SSDS are web services, but utilize interfaces more closely aligned with the REST style of service definition. Our users do not have the resources to hire experienced web service developers or learn SOAP, but were very interested in the lower overhead protocols like REST.

Related to the importance of interfaces is the importance of defining which requirements will not be addressed by the interfaces and components and where policies (and their enforcement mechanisms) must be defined. Software cannot meet every system requirement and often policies have to be defined to keep the development tractable. These policies should be as rigidly defined as the interfaces and go through the same change procedure as the defined interfaces.

As part of the interface contracts, we also found it crucial to identify the metadata that was critical and would have to be robust in the system. Normally interface definitions state hardware and/or software interaction interfaces, but do not specify the language that is to be spoken across those interfaces. In the SSDS, for example, we use a messaging system to communicate with SIAM, and although that interface was fairly straightforward to define, the actual language and structure of what was in the message occasionally was not as rigorously defined. We found at times that one component relied on the robustness of certain metadata, while another did not have such rigid requirements. This metadata should be defined in the interface specification and subject to its change control process.

In addition to defining the key metadata, the various components should consume that metadata as early as possible in the development process to identify issues and necessary design changes. Simulations are a necessary step in developing the system, but the different components should try to utilize the interfaces (both software and metadata) as soon as realistically possible. Also, we found that some metadata should not be statically defined in the instrument, but should be generated on the fly by the instrument (or instrument service). This minimizes stale metadata where static metadata was not kept in synch with changes to the instrument or infrastructure.

Once the observatory is in an operational state, we discovered the need for a testing station where an instrument that was soon to be deployed in the observatory could be connected. Having this test harness aware of the state of the observatory, but not able to change it, would have alleviated some issues that arose from incorrectly configured instruments.

Acknowledgment

We would like to thank the entire MOOS development team and its project manager, Keith Raybould. Specific thanks go out to MOOS team members Kent Headley and Mark Chaffey. We thank our always supportive science and operations teams in the mooring and AUV projects, especially John Ryan, Hans Thomas, Erich Rienecker, Duane Thompson, Paul Coenen, and Ken Heller. Of course, thanks also go to the SSDS development team of Andrew Chase, Mike McCann, Brian Schlining, and Rich Schramm. SSDS work is funded by the David and Lucille Packard Foundation.

References

[1] D. Edgington and D. Davis, eds., "SENSORS: Ocean Observing System Instrument Network Infrastructure," *NSF Sensors Workshop*, 2004.

[2] ORION Executive Steering Committee. "Ocean Observatories Initiative Science Plan", Washington, DC, 2005.

[3] Ocean ITI Working Group, "Trends in Information Technology Infrastructure in the Ocean Sciences", 2004.

[4] I. Foster, "Service-Oriented Science", *Science, Volume 308*, pp. 814-816.

[5] "The Integrated, Strategic, Design Plan for the Coastal Ocean Observations Module of the Global Ocean Observing System", *GOOS Report No. 125*, IOC Information Documents Series No. 1183, UNESCO 2003.

[6] Data Management Committee, "Workshop Report", *Data Management for Marine Geology and Geophysics – Tools for Archiving, Analysis, and Visualization*, 2001.

[7] T. O'Reilly, K. Headley, et. al., "MBARI Technology for Self-Configuring Interoperable Ocean Observatories", Oceans 2006 Proceedings, 2006.

[8] D. Edgington, D. Davis, T. O'Reilly, "OCEAN OBSERVATORY Instrument Software Infrastructure", Oceans 2006 Proceedings, 2006.

[9] St. Arnaud, B., A.D. Chave, A. Maffei, E. Lazowska, L. Smarr, and G. Gopalan, An integrated approach to ocean observatory data acquisition/management and infrastructure control using web services, *Mar. Tech. Soc. J.*, *38*, 155-163, 2004.