Enabling Data Sharing with the Shore Side Data System (SSDS): Lessons Learned and Future Development

Kevin J. Gomes, Luis Bermudez, and Karen A. Salamy *(Monterey Bay Aquarium Research Institute, 7700 Sandholdt Road, Moss Landing, CA 95039)*

IN33B-1182

Abstract

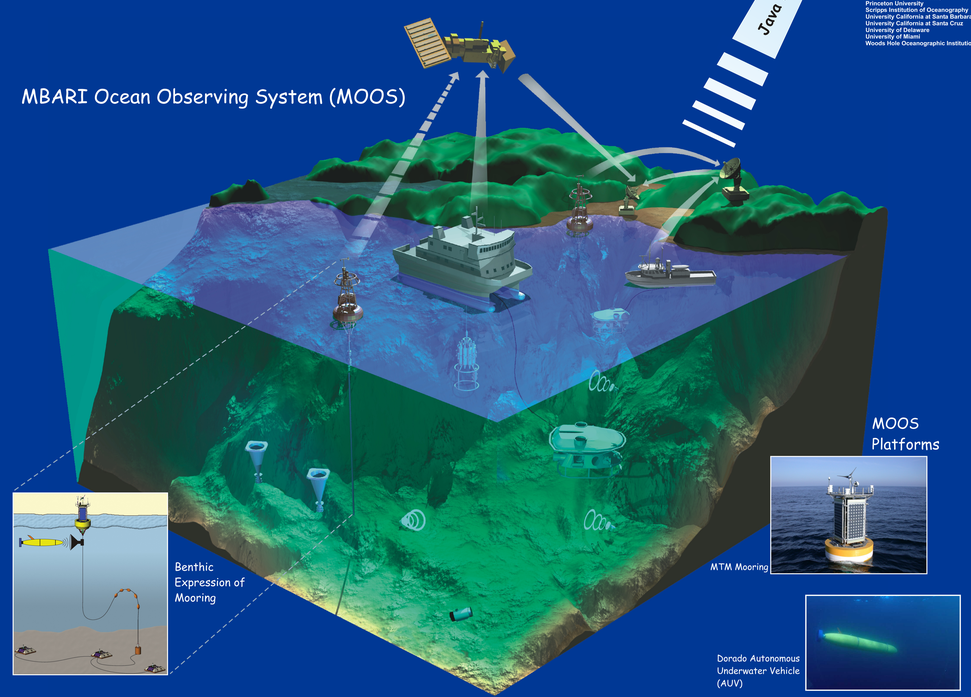
The Monterey Bay Aquarium Research Institute (MBARI) has extensive experience building data systems to meet scientific needs in oceanography. Many of these experiences involve system communication and data interoperability. Obviously there are many issues associated with "systems of systems", and while this is a more tractable problem within the walls of an institute, when interoperating with external data systems the difficulties grow rapidly. One example of a project within MBARI, the Shore-Side Data System (SSDS), has a service-oriented architecture which helped immensely in the connection of systems. The SSDS has served as a portion of the data management system for the Center for Integrated Marine Technologies Wind to Whales Program. Software connectivity was greatly enhanced by the service-oriented architecture of SSDS, but we still faced issues surrounding common vocabularies and ontologies that are necessary for true system interoperability. With the help of the Marine Metadata Initiative (MMI), some promising technologies are being developed to help SSDS bridge the gap between itself and other data systems. This poster presents the SSDS system and the portions of the architecture that have assisted in the connection of some production systems. It also details where we are currently experiencing limitations and how projects like MMI are enabling interoperability with external data systems.

Introduction

MBARI has been developing ocean observatory technology for many years. The Monterey Ocean Observing System (MOOS) program at MBARI has produced many technologies during this development effort and has learned many lessons as well. In addition to the hardware and software development for the actual observatory assets, the SSDS project was developed to manage the data and metadata of the observatory. SSDS manages the infrastructure data and metadata and provides both human and programmatic access to this information. During the development and operation of the SSDS, further issues were uncovered that went beyond the boundaries of a single data management system. One of those issues was naming conventions and standards. MMI has been working on technologies, prototypes and tools to help solve these system impedance mismatches. With these technologies, we have successfully demonstrated, using web services, the interoperation of the SSDS with another data system for the Autonomous Ocean Sampling Network (AOSN) program.

Data and Metadata Management

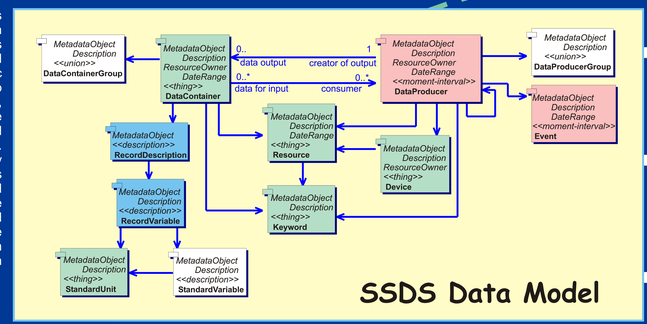
In MOOS, deployed assets such as instruments and vehicles collect data in their own formats and then ship that data back to shore using the underlying infrastructure called Software Infrastructure and Applications for MOOS (SIAM) using the Java Messaging Service (JMS). The SSDS then collects all metadata and data shipped over the SIAM infrastructure as well as that collected from shore-side data processing and user input.



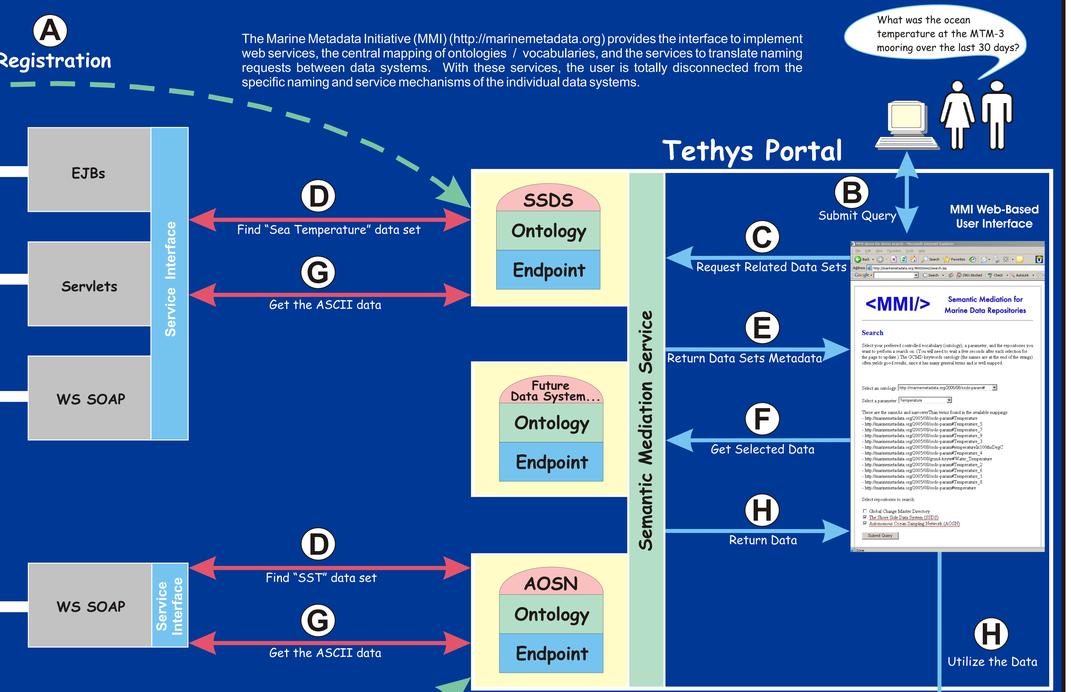
Service-Oriented Access

The metadata is tracked using a persistent data model and serves as the core information repository for users and programs to access data and metadata in the SSDS. In order to effectively give access to this metadata and data, services were then layered on top of the persistent model to both control the business logic surrounding the access points and to create services that any client can call. The SSDS is a Java Enterprise Application (J2EE) that exposes its functionality through Enterprise Java Beans (EJBs), servlets and web services using Simple Object Access Protocol (SOAP).

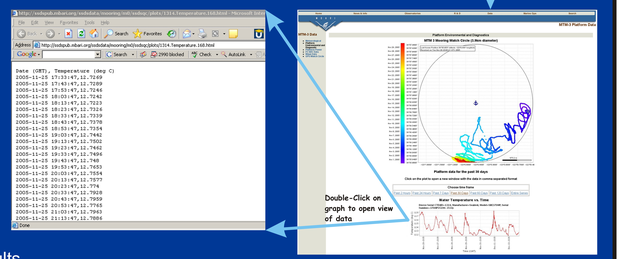
Right: This data model has been very successful in helping the SSDS meet its requirements as a system and meet the needs of the scientific user community at MBARI. To interact with this data model, service methods were developed using EJBs and then exposed as web services. Since a data system can only partially isololate the users from the underlying data and metadata model, the service oriented architecture allowed us to quickly create aggregate services that were key in enabling SSDS to interact with other systems.



Right: AOSN Partners and Collaborators:
California Institute of Technology
California Polytechnic State University
Harvard University
Monterey Bay Aquarium Research Inst.
NOAA. Pacific Fisheries Env.I Lab.
NASA Jet Propulsion Laboratory
Naval Postgraduate School
Naval Research Laboratory
Princeton University
Scripps Institution of Oceanography
University California at Santa Barbara
University California at Santa Cruz
University of Delaware
University of Miami
Woods Hole Oceanographic Institution



The Marine Metadata Initiative (MMI) (http://marinemetadata.org) provides the interface to implement web services, the central mapping of ontologies / vocabularies, and the services to translate naming requests between data systems. With these services, the user is totally disconnected from the specific naming and service mechanisms of the individual data systems.



Steps in User Query of Interoperating Systems

A. Data systems that desire interoperability first create and **register** their ontology mappings and service endpoints with the Tethys Portal, which acts as a service broker. As a result, Tethys has all the information about the various SOAP Web Services that it needs to solve semantic disconnects.

B. The MMI Web-Based User Interface allows a user to **submit a query** to three different project data repositories (SSDS, AOSN, and the Global Change Master Directory) based upon each project's unique terminology (the project's ontologies and data parameters).

C. Once the user submits his / her specific query, MMI looks up each native data system's parameter names using the sameAs and narrowerThan functions to assure that it **requests related data** from each data system in its native dialect.

D. Once the Tethys system finds all matches in the ontology map(s), for the data system(s), it then grabs all the different data system web service endpoints and calls the getMetadata() service for each data system with the mapped parameters. From this it **finds any relevant data sets**.

E. The Tethys system then **returns the metadata** for any data sets found with the requested parameter. The metadata includes the repository and data set IDs, and information characterizing the data set.

F. The user can then choose which query returns are applicable and **get the selected data** sets of interest.

G. The selected data sets are then passed on to the Tethys portal, which again grabs the service endpoints and calls the getASCII() method to **get the ASCII data** from each repository.

H. Tethys then **returns the ASCII data** to the user or the requesting system, which can then **utilize the data** to create data plots and web-pages for the user to view.

Acknowledgements

We would like to thank the David and Lucile Packard Foundation for supporting this work. We would also like to thank the MOOS, SIAM, SSDS, and MMI development teams as well as the AOSN program (specifically Mike Godin) for developing the AOSN web services to make the interoperability possible.

Results

With the MMI/Tethys system in place, we have a critical piece to enable users to have truly interoperable systems. Although it is very early in the development of these systems of systems, we currently have prototypes that allow users to make metadata and data queries that cross over multiple interoperating systems (SSDS/AOSN).

Lessons Learned and Issues

Although the SSDS has been able to meet the requirements for the MOOS program to date, there have been lessons that were learned and issues that surfaced during the operational phase of SSDS. One such issue was that without naming standards and conventions, user access to metadata and data is often incomplete and sometimes fragile. If a user sets up a query to extract information from a system based on a specific variable name, and if someone adds an instrument variable to the system whose name does not match the one requested, then it will not be returned in the query. We found that without conventions, names would often be assigned that had overloaded meaning that users came to depend on. When the name was changed, this dependency caused clients to fail.

Conclusions and Future Work

The successful deployment and operation of the MOOS, SIAM, and SSDS projects has given us a valuable platform on which to test new technologies and gain valuable insight into scientific data management. When exporting data from the MOOS project, we found that users immediately desired simpler ways of collecting data from multiple systems (not just MOOS). With the help of the MMI project, we have implemented a prototype that allows users to query across multiple data systems without having to know the specific naming conventions or programming interfaces of each individual system. The next steps are to expand the set of service capabilities and define the various interfaces that will allow for more capable user queries. We would also like to include more data systems to ensure that the interfaces and services being developed meet the needs of the users and the various system developers.

Compiled by Karen A. Salamy for MBARI, December, 2005