

# An SOA-based Framework for Instrument Management for Large-scale Observing Systems (USArray Case Study)

Corneliu Cotofana<sup>1</sup>, Longjiang Ding<sup>1</sup>, Peter Shin<sup>1</sup>, Sameer Tilak<sup>1</sup>, Tony Fountain<sup>1</sup>, Jennifer Eakins<sup>2</sup>, Frank Vernon<sup>2</sup>

<sup>1</sup>*San Diego Supercomputer Center, University of California San Diego  
9500 Gilman Drive  
La Jolla, CA 92037-0505  
neil@spsc.edu*

<sup>2</sup>*Scripps Institution of Oceanography University of California San Diego,  
9500 Gilman Drive  
La Jolla, CA 92037-0701*

## Abstract

*Large-scale observing systems are poised to become the dominant means of study for a variety of natural phenomena. These systems are comprised of hundreds to thousands of instruments that must be queried, managed, and shared in a scalable fashion. Services-oriented architectures (SOAs) are widely recognized as the preferred framework for building scalable and extensible cyberinfrastructure. By applying SOA concepts, we created a framework for organizing observing system resources. Guided by this framework, we developed web services, custom workflow applications, and an integrated user interface of monitors and controls for managing instruments in large-scale sensor network observing systems. In this paper we present our approach and discuss its application to the NSF EarthScope USArray large-scale seismic observing system.*

## 1. Introduction

One of the most overlooked challenges involved in managing large scale observing systems is the human effort required to configure, deploy, and monitor the thousands of instruments and real-time data streams. For example, the National Science Foundation (NSF) [2] EarthScope USArray project [4], which studies the seismic tomography of the continental United States, includes thousands of sensors being constantly deployed and redeployed. This process required administrators to log in to multiple computers, edit configuration files, and run executables to properly integrate the new equipment into the existing sensor network. In this paper, we describe our application of a services oriented architecture (SOA) approach to the current USArray real time infrastructure to streamline the instrument management process. We developed a layer of command and control web services on top of the existing sensor network middleware used by USArray and built web-

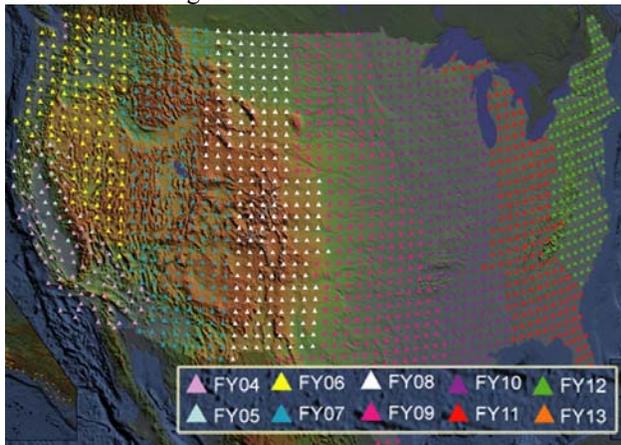
based management applications that orchestrate these services to automate common management tasks. We then added the geographical information system (GIS) capabilities provided by Google Earth™ to display the sites being configured and their surrounding environments. By developing these extensions, we were able to centralize management activities into one consistent interface, decoupled from the actual systems hosting the underlying sensor network middleware. Additionally, the management tools provide a means of keeping track of multiple instrument sites, as they go through the various deployment steps, and automatically check the constraints, reducing input errors. Furthermore, these tools guide the administrators through the configuration of new sites, ensuring that all steps are properly completed in the right order. GIS tools for sensor networks facilitate a number of administration tasks, ranging from visual verification of site coordinates to the planning of new deployments given natural environmental conditions. By integrating all these components, we have created a cyberdashboard (an integrated user interface of monitors and controls) for observing system management. The cyberdashboard provides an intuitive and comprehensive view into system status and operations, as well as control functions over various system resources. This includes data streams, instruments, data collections, and analysis and visualization tools. We believe this to be a useful approach applicable to many other large scale sensor network projects.

## 2. Background

The NSF EarthScope provides a framework for broad, integrated studies across the Earth sciences, and investigates the structure and the evolution of the North American continent [3]. The USArray is a component of EarthScope. It is a continental-scale seismic observatory designed to provide a foundation for integrated studies of

lithosphere and deep Earth structure over a wide range of scales [4]. The USArray consists of thousands of sensing instruments and telemetry hardware which generate thousands of real-time data streams. Given the limited budget of the project and the high cost of the sensing and telemetry equipment, there are not enough sensors to monitor the entire United States at the scale desired. In order to achieve the broad topological coverage envisioned, these sensors must be deployed and redeployed in phases, monitoring one region at a time, for a given period. This repetitious task of deployment and redeployment of equipment over the lifetime of the project could greatly benefit from a customized instrument management cyberdashboard. A cyberdashboard would ease the burden on the sensor network administrators, allowing them to focus on other equally pressing tasks.

The USArray consists of the following two major categories of equipment. The first is a transportable array (TA) of 400 portable, autonomous broadband seismometers deployed on a uniform grid that will systematically cover the United States in segments. The TA will move across the country approximately every 18 months, from west to east, as shown in Figure 1. The second is a flexible component (FA) of 400 more portable broadband seismometers and 2000 single-channel high frequency recorders that are deployed to study particular areas of interest. The majority of these seismometers require separate telemetry units, deployed alongside the sensors, to digitize measurements and provide a network interface to the data generated.



**Figure 1. USArray 12 year continental deployment plan**

The USArray project requires a robust data acquisition and stream management infrastructure. This infrastructure was developed using technology from the Real-time Observatories, Applications, and Data management Network (ROADNet) project [6], and is operated by the Array Network Facility (ANF) [5] located

at Scripps Institution of Oceanography (SIO) [1]. The ANF was formed to ensure the data quality and availability from the aforementioned transportable and flexible seismic sensor arrays. The ROADNet project supports a variety of sensor network activities through a combination of software tools and network communications infrastructure. ROADNet software technology consists primarily of two components: sensor network middleware and data grid [10]. The Antelope Real-Time System (ARTS) [25] sensor network middleware provides the following necessary services: data acquisition, reliable data transport, and a homogenized platform for data access and application development. USArray data archiving, replication, and virtualization at the ANF are achieved through the use of the Storage Resource Broker (SRB) data grid technology [8][9]. The integration of ARTS and the SRB is referred to as the Virtual Object Ring Buffer (VORB) [7].

The USArray team uses a two-phase deployment process for all new monitoring sites to assure data quality and overall system reliability. At the central management facility, there exist two parallel instantiations of the data acquisition software infrastructure, known as the preliminary and production environments. The difference between the preliminary and production environments is the set of users that consume the data streaming in to each. Data from the preliminary environment is used primarily by observing system administrators and technicians for the purpose of configuration, system testing, and data quality assurance/quality control (QA/QC). Data that is streamed through the production environment is meant for actual domain scientists and end users, thus the quality of the data, as well as its timely availability, must be ensured.

New sites are first configured to connect and stream data to the preliminary sensor network middleware node. Here, the middleware is configured to properly communicate with the hardware and to buffer, represent, and archive the data streams from this source. The site is then monitored for any anomalies in sensor configuration or resulting data quality. Once the new site has been deemed properly calibrated and configured, it is switched over to the production sensor network middleware environment. This environment is identical to its preliminary counterpart with the exception that several additional processes operate on the real-time data streams. These processes include long term archiving at the ANF via the SRB data grid technology and analytical tasks such as real-time detection and correlation of interesting events. The Incorporated Research Institutions for Seismology (IRIS) Data Management Center (DMC) [24] subscribes to the seismic data streams from the production environment for the purpose of archiving and dissemination. Domain scientists subscribe directly to

IRIS Data Management Center to acquire data for their research.

Although the basic data acquisition and distribution infrastructure is in place, instrument management activities within the USArray are still complex and time consuming. Not only are there a large number of sites involved, but each site goes through over fifteen deployment steps, including the hand editing of multiple configuration files, execution of command-line programs, and logging into multiple machines. In order to make the management process scale up, we have applied Services Oriented Architecture (SOA) [20] principles to the current USArray infrastructure. SOA has been applied in a variety of industrial and, more recently, scientific [15] domains, to facilitate scalability and interoperability among software system components. The goal of SOA is to achieve a loose coupling among interacting software agents through the use of simple and ubiquitous interfaces. A service can be defined as a unit of work done by a service provider to generate a result on behalf of a service consumer. Typically, web and grid service technologies have been used to implement these service interfaces.

In applying SOA principles to the USArray, we first wrapped the various sensor network middleware resources involved in the deployment process using web services. Exposing common tasks as web services eliminates the need to log in to multiple machines. Furthermore, we were able to abstract middleware-specific syntax and configuration file details behind a high-level, platform-independent XML interface. By making them atomic web services, we were able to reuse them in multiple applications and automate key deployment steps, making them more efficient. Additionally, we were able to include various constraint checks in the web service application logic, potentially reducing the number of configuration errors.

We then orchestrated the services into workflows for the deployment of new sites and other management tasks. These workflows help to organize the steps involved in configuring, testing, and deploying new USArray monitoring sites. Workflows capture the complex logic employed by the system administrators and provide guarantees that processes are followed correctly and efficiently. By making the deployment process an explicit workflow, check-points can be established, and multiple sites, in various stages of deployment, can be more easily organized and managed. Through workflows, management functionality is aggregated into one interface, reducing the actual number of disparate operations the administrator must perform. Since these management workflow applications make use of the web services layer built on top of the sensor network middleware, they do not have to reside on the same

physical machine as the middleware, providing a degree of flexibility in the overall system architecture.

### 3. Related work

Within the domain of sensor networks, there are many different equipment manufacturers, each with their own proprietary software drivers and utilities for properly utilizing the hardware. Often there is little to no interoperability built in to these sensor network software suites, forcing sensor network-based observing system engineers to create custom solutions. The need for a common layer of cyberinfrastructure, especially in large observing systems, with heterogeneous collections of instruments, has been recognized and there are multiple efforts currently working to achieve this [11]. Such a layer will facilitate the development of integrated management and scientific applications by providing a standardized set of tools, APIs, and services that serve as building blocks. One project working to achieve a common middleware layer on top of disparate sensor networking hardware and vendor-specific software is the ROADNet project.

The ROADNet project uses the Antelope Real-time System (ARTS) sensor network middleware to provide a common application development layer over thousands of heterogeneous instruments. Antelope provides reliable data acquisition interfaces, data transport, a network transparent data stream management architecture, and various utilities. Most importantly, Antelope provides a common API layer for basic data and instrument interaction. Antelope provides support for a variety of telemetry instrumentation, as well as a framework for integrating new kinds of sensing equipment. While ROADNet has achieved a robust data acquisition and monitoring system, management functionality is dispersed in multiple locations throughout the software installation. There is much room for the kind of automation required for large-scale observing systems with frequent deployments such as the USArray. At present, all applications built directly on top of Antelope have to use Antelope-specific APIs, executables, and conventions. This makes interoperability, or cross-observatory data and instrument control applications more difficult for those systems that do not use Antelope. Our work has addressed this limitation by exposing key functions of the Antelope system as self-describing, platform-independent web services that can be orchestrated into workflows and custom applications.

Two issues not specifically addressed by the creation of a common application programming interface (API) layer are scalability and interoperability between different observing systems. These issues will be critical to the success of next-generation large-scale sensor network-

based observing systems such as the National Ecological Observatory Network (NEON) [16], Ocean Research Interactive Observatory Networks (ORION) [19], and Collaborative Large-Scale Engineering Analysis Network for Environmental Research (CLEANER) [18]. For these reasons we chose to build our instrument management solution for the USArray using Service Oriented Architecture (SOA) concepts. One notable example of the successful application of SOA to the management of large scale infrastructure is the work done by CANARIE, Canada’s advanced Internet development organization [12]. CANARIE has employed web services and SOA to dynamically control their advanced optical fiber networks.

CANARIE has successfully demonstrated the utility of applying SOA principles to the administration of large-scale infrastructure with the establishment of CA\*net 4 [12], Canada’s next-generation optical fiber network, and the User Controlled Light Path (UCLP) [12] software used to dynamically control it. CA\*net 4 was designed so that all network infrastructure resources, such as optical switches and routers, had web service interfaces. The network was then partitioned into domains which can be managed and controlled by end users. Consumers can use the UCLP software to dynamically create dedicated optical networks (light paths) for specific applications or communities. UCLP does this by creating network management workflows using the infrastructure web services as building blocks. Currently CA\*net 4 is connected to over one hundred colleges and universities, two thousand schools, and various research institutions, and government departments.

#### 4. Our approach

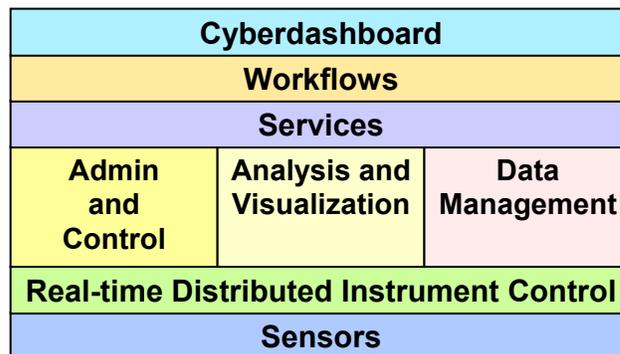
As mentioned in the background section, there are numerous challenges involved in the administration of large-scale sensor networks. This is particularly true of those observing systems, such as the USArray, which feature temporary or rolling instrumentation deployments. Given the complex deployment process and the large number of sites involved, day-to-day operations place an ever increasing burden on administrators. In order to address these challenges, we have applied SOA concepts, similar to those employed by CANARIE, to decouple management functionality from the physical installation and streamline common tasks.

##### 4.1 SOA-based framework for observing systems

Having participated in multiple sensor networking projects [26], including structural health monitoring [26], limnology [13], and seismology, we have observed a set of requirements that is common across domains. We recognize the utility of a generalized framework with

which to organize observing system resources. Having such a framework will facilitate scalable resource management and administration. For large-scale observing system projects such as NEON, ORION, and CLEANER, which have mandated interoperability requirements, a generalized observing system framework will be critical. Starting with the SOA concept of wrapping resources with services, then orchestrating services into applications using workflows, we designed the observing system framework shown in Figure 2. Since we want this framework to have broad applicability in a variety of scientific domains, we designed it to be an open, technology agnostic architecture that does not commit to particular software products. This is also the approach we have used within the USArray as described in this paper.

In this approach, the services provide the base-level building blocks for more complex applications. Workflow tools and custom applications are employed to orchestrate the underlying services and provide an additional level of process automation. By structuring the flow of data and control commands, the overall system is more efficient (fewer human interventions) and safer (fewer opportunities for inappropriate actions or out-of-sequence tasking). Finally, a cyberdashboard provides a secure user interface into system applications that can be personalized for individuals or groups of users (e.g., administrators, engineers, scientists, and the public). This approach provides a modular system architecture that supports interoperability among systems components in an efficient and secure framework.



**Figure 2. SOA-based open architecture for observing systems (and USArray implementing technologies).**

For the USArray case, we implemented the generalized framework using the following technologies. In Figure 2, the bottom layer, sensors, consists primarily of Streckeisen STS-2 and Guralp CMG 3T seismometers, as well as Kinemetrics Q330 data loggers with built-in GPS. The next two layers above: real-time distributed instrument control, as well as admin and control, and analysis and visualization, and data management, are

implemented using the VORB as described in the background section. Our work is focused on the top three layers. The web services are implemented using Java, Apache Axis soap toolkit [23] and Apache Tomcat application server [22]. The workflow layer is implemented using custom Java-based web applications. The cyberdashboard was developed from an integration of Gridsphere [14] and Google Earth™ [21]. The latter components will be discussed in the following section.

## 4.2 Command and control web services

In the process of deploying a new instrument site, USArray administrators need to interact with various components within the Antelope sensor network middleware. In the past, this was a tedious manual process. Deploying a new site required logging in to multiple machines, due to the two-phase deployment architecture utilized by the USArray (see background section). For this reason, we wrapped the key VORB resources using web services, enabling them to become building blocks in multiple applications. By building the services layer on top of the sensor network middleware (administration and control, analysis, data management), we avoided the problem of having to directly interact with multiple, vendor-specific software drivers. In this respect, the sensor network middleware provides a degree of uniformity regardless of the specific instruments that are involved. The suite of web services is then deployed on every node which is also running VORB software within the sensor network.

The following Antelope modules must be manipulated as part of the deployment process for a new monitoring site: `rtexec` (tasking and control), `q3302orb` (instrument communications), `dbbuild/dbmerge` (instrument metadata generation and cataloging), `ttgrid` (seismic analytical process), and `orbmonrtd` (waveform data visualization). For each module, a web service was created, with multiple methods, so that it could be utilized as part of one or more administrative workflows. Each of these sensor network middleware modules is a command-line executable. In some cases, the web service invokes this executable directly with a specific set of arguments. In other cases, when the module runs continuously, its corresponding configuration file is modified using the Java API we have developed for parsing Antelope configuration files (`pf2java`). The updated module is then restarted using the `rtexec` module on the same system. The web services created for the USArray to implement the SOA-based management solution are detailed below.

1. `Rtexec` web service: tasking and control. This service is used to define and toggle the execution of tasks and modules within the Antelope sensor network middleware. Almost all other functions within Antelope, including administrative, analytical, and

data-related processes, exist as tasks defined within the `rtexec` module. The `rtexec` web service is used to remotely start, stop or reconfigure these tasks.

2. `Q3302orb` web service: instrument communication. This service is used to register new instrumentation into the Antelope sensor network middleware, and initiate communication with the equipment. The web service interacts with the `q3302orb` module using `pf2java` and the module's configuration file.
3. `Dbbuild/dbmerge` web service: instrument metadata generation and cataloging. Each new instrument connected to the sensor network middleware will generate multiple channels of real-time data. This service creates the metadata files used by the Antelope middleware to correctly identify and interpret the real-time data streams. The `dbmerge` method of this web service will then catalog the metadata into Antelope's internal registry, making it available to other modules and processes. `Dbbuild` and `dbmerge` are command-line executables within Antelope.
4. `Ttgrid` web service: seismic analytical process. The USArray is constantly monitoring seismic activity. Analytical processes for earthquake event detection, correlation, and notification run continuously on incoming streams of real-time data. The `ttgrid` module plays a key role in this process and must be updated to incorporate data sources from newly deployed monitoring sites. The `ttgrid` web service accomplishes this task by first generating the new configuration files used by the related analytical processes then restarting them via the `rtexec` web service.
5. `Orbmonrtd` web service: waveform data visualization. USArray administrators use the `orbmonrtd` Antelope module to visualize incoming waveform data continuously for the purpose of troubleshooting data and instruments. Whenever a new instrument is added to the system, `orbmonrtd` must be updated to incorporate the new data streams. The `orbmonrtd` web service updates the corresponding module by manipulating its configuration file using the `pf2java` API mentioned above.

The web services we have created on top of the VORB to implement the services layer are written in Java. The Apache Axis 1.1 Soap toolkit is used to create and parse the Soap messages used to communicate with the web services. The Apache Tomcat 4.1 application server is used to host the Soap server and web services. For maximum interoperability, Axis "message literal" web services have been used. Axis message literal web services transmit the XML payload of the web services messages un-encoded, as a string. This allows the consumption of our web services by future clients while avoiding the encoding inconsistencies present between different implementations of the Soap message library. A

Web Service Description Language (WSDL) [28] descriptor file is provided for the generation of custom web services clients.

### 4.3 USArray instrument management applications and workflow

After creating a layer of services on top of core observing system resources and functionality, we developed applications for key observing system management tasks. In particular, the complex process of deploying a new monitoring site has been made into an interactive workflow application, called the Site Deployment Wizard, which guides operators through each step. To handle the large number of sites being deployed at any one time, a Site Manager application was developed to keep track of site-specific information. The Site Manager also allows administrators to change parameters, stop and resume the deployment process at any step. The Site Manager and Site Deployment Wizard exist as web-based applications, accessible from any web browser. They are decoupled from the underlying sensor network middleware. These applications are federated into a web portal using the GridSphere portal toolkit [14]. Access is regulated using GridSphere's support for user roles and login-based accounts. Additionally, a cyberdashboard for the USArray observing system has been developed by integrating the previously mentioned management applications with Google Earth™. All the applications are discussed in more detail later in this section.

In turning the manual site deployment process into a workflow, we first formally defined each step that the new site must go through. Defining the deployment steps is necessary to ensure that they are done in the proper order and that no step is accidentally omitted. Furthermore, making the deployment steps explicit facilitates the creation of logical checkpoints at which to stop and resume the deployment of each site. Formalizing the administration tasks as workflows also enables us to identify the specific pieces of information required at each step, as well as introduce constraint checks to verify the validity of this information before executing that step. We generated over fifteen different steps in all, starting with the initial notification from the field engineer after the instrumentation has been physically installed, to the point at which it is fully operational within the production USArray environment and generating useful scientific data.

Having defined the workflow for deploying a new site, we then created a software model to implement this workflow. The model enforces predefined constraints on the site-specific information required for each step and ensures proper execution order of all deployment steps.

This model is used both by the Site Deployment Wizard and Site Manager.

The Site Deployment Wizard uses the model described above to guide the administrator through the instrument deployment workflow steps, invoking the appropriate web services and reporting the results. A description is given of the current step, as well as a list of all the steps involved, with the current step highlighted. The execution output is displayed in the results view, as shown in the screenshot in Figure 3, along with the option to re-execute the current step. The administrator is prevented from advancing to the next step until the previous has completed successfully. The set of parameters associated with the particular site being deployed is also displayed. The Site Deployment Wizard allows the administrator to revisit and re-execute any prior site deployment step. This is especially useful if some parameter needs to be updated or a given step had failed to execute properly the first time. The Site Deployment Wizard provides a single interface to configuration steps that previously required interacting directly with many different components within the VORB sensor network middleware.

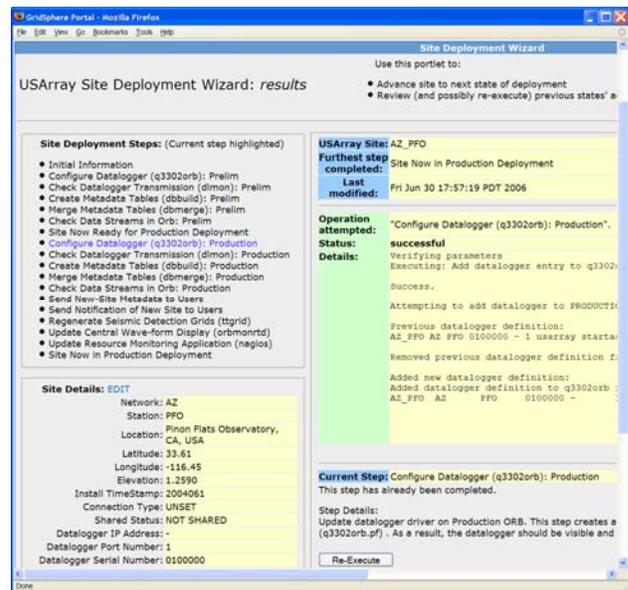


Figure 3. Site Deployment Wizard screenshot.

The Site Manager is designed to assist administrators in state-of-health monitoring and observing system management. It greatly simplifies the task of keeping track of the large number of instrumentation sites deployed at any given time. For the USArray, as many as 20 new sites are installed weekly by field engineers. These sites must be brought online and integrated into the observing system infrastructure as quickly as possible. Given the numerous steps involved in site deployment, it

can be difficult for administrators to keep track of the status of each site. Using the Site Manager, an administrator can quickly see which and how many sites are currently in the process of being deployed, and the current status of each (see Figure 4). This way, she can ensure that no site is overlooked or falling behind schedule. Additionally, the Site Manager allows site-specific information such as communication parameters or geospatial coordinates to be viewed and updated. Although most of this information is present within the underlying sensor network middleware, it is scattered in various database tables and configuration files. The Site Manager also utilizes the software model of the site deployment workflow to automatically re-execute all steps that depend upon the updated site parameters. In this way, the representation of the site within the Site Manager remains synchronized with its instantiation in various places within the VORB sensor network middleware. At any point, the Site Deployment Wizard can be invoked to resume the deployment process of a particular site. The Site Manager uses a relational database to store information about each monitoring site. Together, the Site Deployment Wizard and Site Manager provide a simple, centralized management interface that abstracts the complex details of the underlying sensor network middleware.

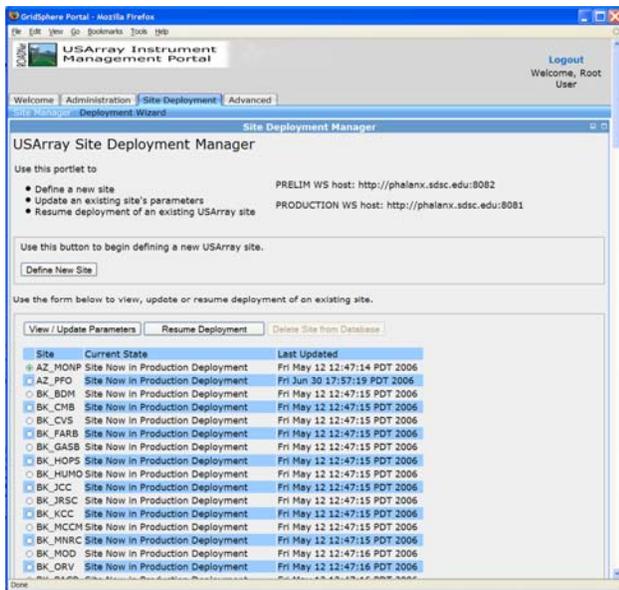


Figure 4. Site Manager screenshot

In managing large-scale sensor deployments, USArray administrators have expressed the need for integration with geographical information systems for visually checking the overall deployment status and the accuracy of geospatial coordinates for specific sites. For the USArray, we have employed Google Earth™ to plot deployed monitoring sites. We have created a

cyberdashboard for the USArray observing system by combining Google Earth™ with the site deployment manager and wizard applications discussed above. The cyberdashboard provides the administrator with an intuitive and comprehensive view into system status and operations, as well as control functions over various system resources. The Google Earth™ interface and the Site Configuration Wizard are integrated to support bi-directional control: user actions on the Google Earth™ interface invoke the Wizard and actions in the Wizard drive the Google Earth™ display. Figure 5 shows a screenshot of the cyberdashboard. The left panel is Google Earth™'s map viewer, which provides a geospatial visual interface into sensor sites. Metadata on sensors and sites are dynamically displayed as the user navigates across the landscape. By clicking on a site, the administrator can quickly view or manipulate the site's configuration parameters using the Site Manager in Google Earth™'s embedded web browser (right panel in Figure 5). Various types of supplemental information can be accessed and integrated into the cyberdashboard displays, including overlays of land cover, utilities, political boundaries, and roads/highways. The cyberdashboard's GIS capabilities provide an intuitive interface that allows users to quickly grasp environmental features of specific sites. The cyberdashboard can also be used to plan the locations and schedules of future site deployments.

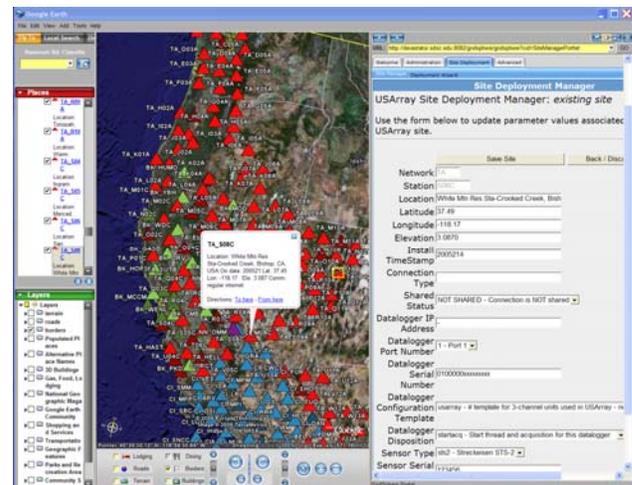


Figure 5. Observing System Cyberdashboard screenshot.

## 5. Conclusion

In this paper, we described our experience in developing a scalable management solution for the EarthScope USArray large-scale seismic observatory. First we created an open architecture for observing systems according SOA principles. This architecture can

be used to organize resources and functionality within observing systems. We implemented the services, workflows, and cyberdashboard layers defined within our architecture. The cyberdashboard facilitates the job of observing system administrators, especially for complex tasks such as the deployment of new monitoring sites. The web services we have built on top of the VORB provide a platform-independent interface layer to instrument management functionality. By orchestrating the services into custom management workflows, we streamlined the deployment process and reduced the opportunities for administrative errors. In doing so, we provided a strategy for dealing with the complexity and the human labor costs involved in managing an observing system of hundreds or more instruments.

The challenges faced by the USArray project, with regard to the scale of their instrument deployments, are not unique to the seismic domain. For other upcoming large-scale sensor network projects, scalability and interoperability are core requirements. In particular, the NEON project, which deals with ecology, and ORION, in the domain of oceanography, will require management and monitoring of thousands of sensors. We believe that our experiences in developing the USArray instrument management system will be valuable in designing the cyberinfrastructure for those projects.

The USArray system provides a proof of concept demonstration of the feasibility and utility of managing observing system operations within a single domain. We are currently expanding this approach to address issues in federated cross-observatory applications. As additional environmental observing systems come into production, it will be important to ensure interoperability. Working across distinct jurisdictional boundaries, with disparate policies and infrastructure, raises significant challenges in application development and system operations. Our current research focus is to understand these challenges and to create the robust cyberinfrastructure necessary for successful pan-observatory applications.

## 6. References

- [1] Scripps Institution of Oceanography. Accessed on 30 Jan 2005 <<http://sio.ucsd.edu/>>.
- [2] National Science Foundation. Accessed on 30 Jan 2005 <<http://nsf.gov/>>.
- [3] EarthScope. Accessed on 30 Jan 2005 <<http://www.earthscope.org/>>.
- [4] USArray. Accessed on 30 Jan 2005 <<http://www.earthscope.org/usarray/>>.
- [5] USArray Array Network Facility. Accessed on 30 Jan 2005 <<http://anf.ucsd.edu/>>.
- [6] Real-time Observatories, Applications, and Data Management Network. Accessed on 30 Jan 2005 <<http://roadnet.ucsd.edu/>>.
- [7] Vernon, F., T. Hansen, K. Lindquist, B. Ludäscher, J. Orcutt, A. Rajasekar (2003). ROADNET: A Real-time Data Aware System for Earth, Oceanographic, and Environmental Applications. *Eos Trans. AGU*, 84 (46), Fall Meet. Suppl., Abstract U21A-06.
- [8] Wan M., A. Rajasekar, and W. Schroeder, "An Overview of the SRB 3.0: the Federated MCAT." Sept. 2003. *SDSC SRB*. Accessed on 30 Jan 2005 <<http://www.sdsc.edu/srb/FedMcat.html>>.
- [9] The Storage Resource Broker. Accessed on 30 Jan 2005 <<http://www.sdsc.edu/srb/>>.
- [10] Moore, R., C. Baru, "Virtualization Services for Data Grids", Book chapter in "Grid Computing: Making the Global Infrastructure a Reality", John Wiley & Sons Ltd, 2003.
- [11] Laboratory for the Ocean Observatory Knowledge INtegration Grid. Accessed on 30 Jan 2005 <<http://lookingtosea.org/>>.
- [12] CANARIE User-Controlled LightPaths. CANARIE. Accessed on 30 Jan 2005 <<http://www.canarie.ca/canet4/uclp/>>.
- [13] Collaborative Lake Metabolism Project. Accessed on 30 Jan 2005 <<http://lakemetabolism.org/>>.
- [14] GridSphere. Accessed on 30 Jan 2005 <<http://www.gridsphere.org/>>.
- [15] I. Foster, "Services-Oriented Science", *Science*, AAAS, Washington, DC, 6 May 2005, Vol. 308. no. 5723, pp. 814-817
- [16] National Ecological Observatory Network. Accessed on 30 Jan 2005 <<http://www.neoninc.org/>>.
- [17] The Consortium of Universities for the Advancement of Hydrologic Science. Accessed on 30 Jan 2005 <<http://www.cuahsi.org/>>.
- [18] Collaborative Large-Scale Engineering Analysis Network for Environmental Research. Accessed on 30 Jan 2005 <<http://cleaner.nacse.org/>>.
- [19] The Ocean Research Interactive Observatory Networks. Accessed on 30 Jan 2005 <<http://www.orionprogram.org/>>.
- [20] He, H. "What is Service-Oriented Architecture." *O'Reilly.Webservices.xml.com*. 30 September 2003. O'Reilly. Accessed on 30 Jan 2005 <<http://webservices.xml.com/pub/a/ws/2003/09/30/soa.html>>.
- [21] Google Earth. Accessed on 30 Jan 2005 <<http://earth.google.com/>>.
- [22] Apache Tomcat. Accessed on 30 Jan 2005 <<http://tomcat.apache.org/>>.
- [23] Apache Axis. Accessed on 30 Jan 2005 <<http://ws.apache.org/axis/>>.
- [24] Incorporated Research institutions for Seismology. Accessed on 30 Jan 2005 <<http://www.iris.edu/>>.
- [25] Antelope Real-time System. Accessed on 30 Jan 2005 <<http://brtt.com/software.html>>.
- [26] Structural Health Monitoring. Accessed on 30 Jan 2005 <<http://healthmonitoring.ucsd.edu/>>.
- [27] Cyberinfrastructure Laboratory for Environmental Observing Systems. Accessed on 30 Jan 2005 <<http://scirad.sdsc.edu/datatech/cleos.html>>.
- [28] Web Service Description Language. Accessed on 30 Jan 2005 <<http://www.w3.org/TR/wsd/>>.