# The SURA Coastal Ocean Observing and Prediction Program (SCOOP) Service-Oriented Architecture

Philip Bogden

SURA, Washington, D.C./GoMOOS, Portland, ME

Gabrielle Allen, Greg Stone, and Jon MacLaren

Louisiana State University, Baton Rouge, LA

Gerald Creager, Larry Flournoy and Wei Zhao

Texas A&M University, College Station, TX

Hans Graber

University of Miami, Miami, FL

Sara Graves and Helen Conover

University of Alabama, Huntsville, AL

Rick Luettich

University of North Carolina, Chapel Hill

Will Perrie

Bedford Institute of Oceanography, Nova Scotia, Canada

Lavanya Ramakrishnan and Dan Reed

Renaissance Computing Institute, Chapel Hill, NC

Peter Sheng

University of Florida, Gainesville, FL

Harry Wang

Virginia Institute of Marine Science, Gloucester Point, VA

***Abstract*-The Southeastern Universities Research Association (SURA) Coastal Ocean Observing and Prediction Program (SCOOP) is a multi-institution collaboration whose partners are working to implement a modular, distributed system for real-time prediction and visualization of the impacts of extreme atmospheric events, including storm surge and wind-driven waves. SCOOP Program partners are developing an interoperable network of modularized components (numerical models, information catalogs, distributed archives, computing resources and network infrastructure) linked by standardized interfaces. This service-oriented architecture (SOA) is emerging as a prototype open access, distributed virtual laboratory for oceanographic research and coastal applications. The SOA approach allows data integration from multiple platforms and enables the exchange of resources, tools, and ideas among a virtual community. The SOA framework consists of five layers: 1) a user interface; 2) an application and tools layer; 3) a management layer; 4) a resource access layer; and 5) physical resources all linked by cross-cutting services. The SOA layer components support several different use cases because they can be configured into a variety of workflows.**

## I. Introduction

The Southeastern Univervisities Research Association (SURA) Coastal Ocean Observing and Prediction Program (SCOOP) (http://scoop.sura.org/) is a multi-institution collaboration whose mission is to create a distributed network of shared resources that will broaden access to the requisite data, models, computational resources, and other key components of a real-time environmental prediction system. SCOOP partners are implementing community information services and technologies to advance the sciences of environmental prediction and hazard planning for our nation's

1-4244-0115-1/06/$20.00 ©2006 IEEE

coasts. The first step toward achieving this vision is to help integrate diverse data flows from a variety of ocean observing initiatives. The second step is to incorporate these data flows into an open-access, scalable, modular, and distributed real-time environmental prediction system. This paper describes the service oriented architecture (SOA) underlying this community resource.

The concept of an SOA is an outgrowth of early advances in information technology (IT) and is a framework for future evolution of the World Wide Web. The SOA approach is recognized by the Ocean.US Data Management and Communications (DMAC) Steering Team as a framework for creating a functional Integrated Ocean Observing System (IOOS). The SCOOP SOA is the result of the combined effort among a team of coastal and computer scientists working for nearly three years to develop and implement the system. Since 2003, the SCOOP Program via its partnering institutions has been incrementally implementing an SOA that provides a cost-effective framework for broad collaboration that can be utilized for coastal research and applications.

## II. Design requirements

Design requirements for the SCOOP SOA are developed by considering a range of possible use cases. For example, extreme events trigger ensemble modeling calculations for the immediate needs of real-time ensemble prediction. The same system supports continuous forecasts for use in day-to-day maritime operations. In addition, a retrospective analysis mode enabled by data archives supports coastal research on past events. A development and testing mode supports research, simulation and innovation of operational capabilities (e.g., multidisciplinary inundation). These multi-mode requirements affect all aspects of the SOA, from computational resource scheduling to data archive, access, and transport.

The SCOOP SOA is composed of a collection of modular components, each providing a well-defined functionality and communicating with the other components across standardized interfaces. As with most modern, distributed systems, the SCOOP architecture relies heavily on web-service interfaces to manage secure resources and data flows across the relatively insecure Internet.

The approach of modularizing components and standardizing interfaces allows innovation by allowing system components to be updated or replaced incrementally in a “plug-and-play” fashion, without impacting other components or overall system operation. With this approach, operational systems can become the focus of ongoing research and development by teams of coastal and computer scientists working together on a common architecture that meets the needs of a broad user community.

Individual components of the SCOOP SOA modularize the functionality common to all of the use cases. Modularization facilitates the subsequent organization and management of the components in a variety of ways to support the use cases. These components include:

- coastal models that predict phenomena such as storrn surge, wind-driven waves, and inundation;
- visualization tools that facilitate efficient analysis of products;
- user friendly interfaces;
- data access, management and catalog services for input and output of data or model comparisons;
- translation and transport services that assure compatibility between the various data flows;
- computing resources that can be organized for quick turn around of large jobs; and
- an active archive of current and historical data and model results for storage, documentation, and retrieval.

## III. Architecture overview

Fig. 1 is a layer diagram providing a hierarchical framework for describing the basic functions in the SCOOP SOA. Each of the five layers comprises a set of components that connect to adjacent layers. Use cases determine the path of the connections between components.

The uppermost *user inteface layer,* interacts directly with the end user. The components consist of a portal for resource access, workflow interfaces, interactive search services, visualization tools, and software libraries of data and models. Components are customized to workflows for specific use cases.

The *application and tools layer* contains the numerical models that predict and analyze environmental phenomena. The components in this layer include data translation and visualization toolkits and the actual workflow configuration tools. Interaction of the components in this layer is coordinated by the workflow tools.

The *management layer* enables workflows by coordinating components in the *applications and tools layer* and by coordinating physical resources in the *resource access layer.* Management services may vary from one use case to another, therefore it is the most flexible in order to accomodate multiple use cases with different workflows.

The *resource access layer* provides standard protocols that connect the *management layer* to physical components for computing, storage, and network connectivity. Resource Access components include middleware protocols such as

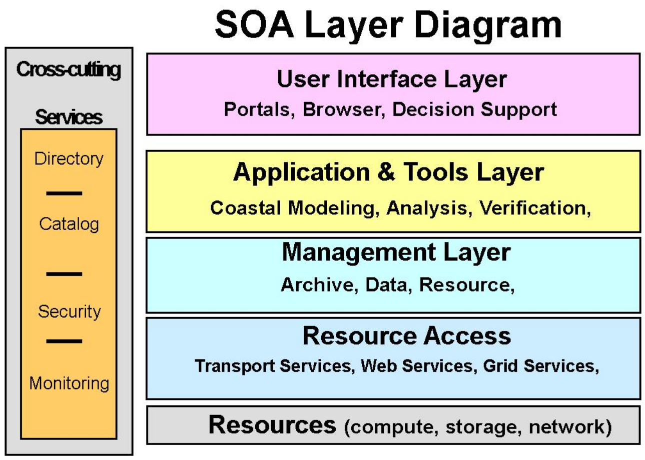
SOA Layer Diagram
Cross-cutting Services
Directory
Catalog
Security
Monitoring
User Interface Layer
Portals, Browser, Decision Support
Application & Tools Layer
Coastal Modeling, Analysis, Verification,
Management Layer
Archive, Data, Resource,
Resource Access
Transport Services, Web Services, Grid Services,
Resources (compute, storage, network)

Figure 1: SCOOP SOA Layer Diagram

Globus and Web Services Resource Framework (WSRF), web-service and data transport protocols such as Simple Object Transfer Protocol (SOAP) and the Local Data Manager (LDM), lower-level networking protocols such as Transmission Control Protocol/Internet Protocol (TCP/IP) and Hypertext Transfer Protocol (HTTP), and virtualization.

The *resources layer* includes all of the physical resources for computing, storage, and network connectivity distributed over the Internet. This introduces special challenges in security and between heterogeneous platforms or operating systems.

Finally, cross-cutting services that interact with multiple layers provide ancillary services that enable *management layer* tasks. The components consist of directories, a catalog, and security and monitoring services.

## IV. Layer Details of the SOA

*The User Interface Layer*

The *user-interface layer* serves as the point of access for models, computing resources, data, and archives. The current user-interface is located at www.openioos.org. The site relies heavily on Open Geospatial Consortium (OGC) standards allowing tools that are being developed to readily "plug into" the SCOOP product streams. It is intended that the OpenIOOS site will serve as a collaborative platform for scientists to share data and models via user accounts, certificates, and policies that are authorization-based. Tools are being developed for workflow orchestration, data and information searches, and data visualization that can be customized for targeted users (e.g., scientists, students, decision makers).

*Application and Tools Layer*

*A. Models and Analysis Tools*

The core components of the *Application and Tools Layer* are numerical prediction models and analysis software. The models currently being displayed on www.openioos.org are Wave Watch III and ADCIRC (See *http://adcirc.org*), wave models and circulation and storm surge models respectively. Additional models may be retrofitted to conform to standardized interfaces and data translation capabilities for input and output data streams. Thus, it will be possible to interchange multiple models in a "plug-and-play" mode. The goal is promoting open access to community models and model intercomparisons potentially advancing ocean modeling science.

*B. Data Translation Services*

Data translation services enable the exchange of data between models with minimal impact on the model code. To make this work, data translation and filter services operate as data are transported between modeling applications. Services include format conversion, region and parameter subsets, point extraction, and re-gridding. Data translation is available regardless of whether the point of origin is external to SCOOP or archived within the SCOOP catalog. Data translation services are being configured for interactive workflows so that users can select from a list of data products, select from a list of translation functions, and determine whether the output is available through the SCOOP transport system or from a repository location.

*C. Visualization*

Users want data and modeling results made available for their favorite analysis tools. Therefore the SCOOP architecture includes standardized interfaces and services for visualizing SCOOP products using a variety of OGC-specified decision support tools. The *www.openioos.org* user interface provides examples of OGC-compliant hurricane data and model results. There are multiple options for users to select data sources, real-time and retrospective predictions, buoy and model data comparisons, etc. The SCOOP program will continue its emphasis on Web-based mapping standards and Geographic Information Systems (GIS) that permit high resolution digital elevation models (DEMs), land use/land cover databases, and other means of displaying street-level resolution.

*Management Layer*

*A. Application and Resource Management*

Managing heterogeneous application environments with their different kernels, distributions, libraries, packages, and configurations is a huge challenge. As with the resource, data, archive, and workflow management components, *applications management* uses protocols from the *resource access layer* to manage flows to and from physical resources. SCOOP's application management relies on portable engineering applications and directory services to address heterogeneity by selecting compatible options for specific applications among all possible SCOOP resources rather than standardizing on a single system configuration.

Resource management components coordinate the physical and virtual distributed resources that support applications run on the SCOOP system. The resource manager is critically dependent on crosscutting monitoring component services that enable dynamic scheduling decisions, which can impact system performance and reliability. Directory services provide dynamic load information, based on the characteristics of individual machines and application runs. The availability of resources is particularly important for ensemble modeling where the completion time is critical.

Management techniques include:

- replicating model runs to ensure timely completion;
- prioritizing individual runs and allocating appropriate resources;
- dynamically acquiring resources as needed (e.g., from SURAgrid or TeraGrid);
- optimizing data movement between resources; and
- setting resource policies for coupled modeling.

*B. Data and Archive Management*

Data management encompasses a range of services required to coordinate observations and model input and output. These data reside within SCOOP archives, the OpenIOOS database cache, and with many external data providers.

The SCOOP archives provide the backbone for accessing a broad collection of observational data and model runs by establishing dynamic and standardized connections to existing repositories where they reside as well as to products created by SCOOP partners. This enables comparisons of observations in real-time and retrospective calculations as well as comparisons between models. The archives hold limited file-level metadata supplemented by catalog services. There are alternative methods, (e.g., LDM and other File Transfer Protocols (FTP) such as Grid FTP), for accessing files in the inventory in order to meet the needs of the user.

*Resource Access Layer*

The *resource access layer* provides protocols for accessing computer, storage, and network resources via the data transport, web services, and virtualization functions. Some of the technologies in this layer are well developed according to stable standards, while others are in a state of development. New technologies implemented in this layer need not affect the functioning of the overlying layers but will realize their benefit.

Data transport protocols allow *management components* to retrieve and/or archive data using SCOOP storage and network resources. There are several mechanisms for data transport: the traditional FTP that moves files from one location to another; Grid FTP as a high performance and secure alternative; and OPeNDAP technology for server-side subsampling as an economic alternative to transferring large files.

Due to the requirements for real-time prediction, data files are pushed via LDM. The LDM has the advantage of moving large files quickly, then triggering subsequent workflows.

Web services provide a flexible mechanism for Application Programming Interfaces (API) and services. Most services conform to World Wide Web Consortium (W3C) standards. Unlike LDM, web services typically employ "data-pull" streams, which are adequate for retrospective analysis. Standards such as SOAP simplify the creation and maintenance of APIs for use in connecting components and managing resources. Protocols developed by the OGC offer geographic content valuable for visualization and data-transport services in coastal operations, especially useful for www.openioos.org.

Virtualization is a relatively new approach to resource access currently being investigated by the grid community (e.g., Globus and the Global Grid Forum) and SCOOP. Virtualization protocols help cope with the problems that may arise when users want to use resources from a diverse set of providers. In essence, virtual machine (VM) environments are configured independently from the software of a physical resource. This layer of isolation from the physical resources can also enhance security.

The virtualization strategy allows resource management components to request virtual resources through a VM management service. This service takes the specification of a VM's configuration and allows for the creation, configuration, and termination of VM instances at run time. While virtualization does impose overhead on the execution time of applications, it is typically small (~ 5 percent) for CPU-intensive programs.

*Resources Layer*

The resources layer represents the physical resources for computation, networking, and storage that are shared in the community and are also an operational tool.

Over half a dozen SURA member institutions are using the SCOOP SOA to utilize new technologies and to advance the science of coastal modeling and prediction. The intent is to leverage established Grid activities such as SURAgrid, the Open Science Grid (www.OpenScienceGrid.org), and others for the ocean sciences, and to increase networking capabilities, perhaps through networks such as the National Lambda Rail, Internet2, or the Louisiana Optical Network Initiative.

*Cross Cutting Components*

*A.. Catalog & Directory*

The directory is a crosscutting service component that enables the user to discover and locate data and resources throughout the system layers. As a primary tool of the directory, the SCOOP Catalog's relational database provides for information-sharing among SCOOP users. The Catalog maintains an inventory of distributed data files of observations and model results that are updated with each run. It provides a targeted location for record retrieval in the SCOOP Archives or links to the data provider source external to the SCOOP architecture. The Catalog provides high-level descriptions of SCOOP data collections, file name conventions, and metadata necessary to document model runs. In addition, it provides the information needed to access particular data streams and the general characteristics and configuration parameters for SCOOP-enabled models,

Additional directory services are currently under development to support automated discovery of available computational resources, maintain shared information on authorized SCOOP users (e.g. grid certificates), and enable distributed shared capabilities. The intention is to allow SCOOP workflows to leverage data and model products in other discipline-specific projects.

*B. Monitoring*

The multi-layer monitoring services support management tasks at all levels of the architecture stack. Thus, for example, there will be prompt detection of overloaded or failed resources during application runs. This allows resource-allocation management so that jobs can be initiated on an underutilized part of the linked computer-resource system. In addition, multi-layer monitoring allows system components to respond to changing environmental conditions. For example, if a hurricane were to change track a different set of prediction tools could be used.

*C. Security*

Methods for ensuring data sharing among multiple institutions according to community standards are under development. Various technologies such as the Globus Grid Security Infrastructure (GSI), firewalls and access control lists are used as appropriate to enable a secure distributed computing environment. Security policies will be in compliance with federal policy and specifications as standards emerge and software becomes available.

## V. SCOOP USE-CASE SCENARIOS in an SOA

Figs. 2 and 3 illustrate in simple terms the system configuration for two SCOOP user scenarios: 1) real-time ensemble prediction and 2) retrospective analyses.

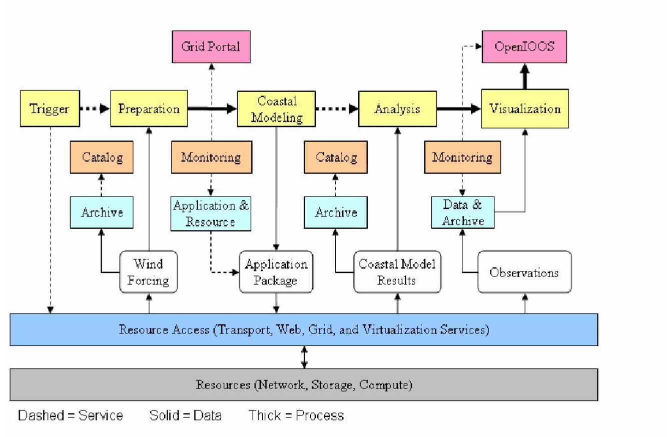

Figure 2: Workflow for real-time event-driven synthetic ensemble prediction visualized on www.openioos.org

prediction. This workflow is triggered by the issuance of a National Hurricane Center warning. The warning initiates the creation of a synthetic wind ensemble. The wind data enter the workflow through the *resource access layer* and are pushed into the *transport layer*. Monitoring and workflow services ensure that winds pass through a translation filter so they can be used by the storm surge and wave models.

Simultaneously, the wind ensemble is placed on the archive and registered in the SCOOP catalog. The resource selection in the *management layer* identifies and selects the available computational resources and launches the model runs across those distributed resources. Meanwhile, observational data are obtained from the appropriate source locations and/or archive. As model results are generated, they are "pushed" to the archives and to verification and analysis tools.

Verification services filter the model results so they can be compared with observational data. Finally, visualization services are used to display the model results and model-observation comparisons on the OpenIOOS website.

Fig. 3 shows the workflow for the retrospective analysis. In this scenario, the sequence of events is triggered through the *user interface layer*. The workflow tool and the *management layer* determine the detailed sequence of actions and component interactions.

Wind fields are obtained from an archive service that interacts with catalog services to locate and retrieve the historical fields via the *resource access layer*. At each stage, transport services initiate interactions between components, which are implemented with web services and supported by data transfer. The translation service delivers the winds in the form required by each of the coastal model components.

For each model run, there is a resource selection process. Once the available resources are obtained, the data and applications are staged at the appropriate locations. Transport moves the model results to the archives and to the verification and validation components. The remainder of the retrospective sequence follows the flow of the real-time ensemble case; i.e., catalog and archive services are used to deliver the observational data for analysis and verification of model results.

The verification/validation services extract data points from model results at locations corresponding to observation data points; both datasets are delivered to the analysis application

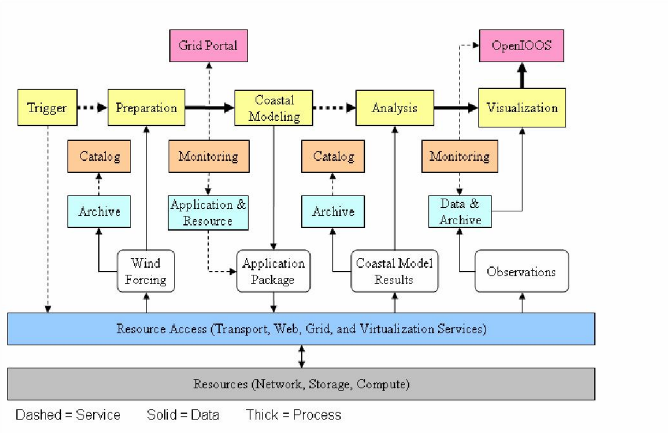

Figure 3: Retrospective analysis initiated by research users

## VI. Conclusions

The SCOOP SOA is being implemented incrementally. Continued development and implementation is based on the most recent benchmark. This incremental approach mimics the so-called spiral implementation model used for enterprise systems engineering in industry.

Another critical and overarching design principle involves adoption and implementation of open community standards. This is critical in making certain there is an effective partnership between the academic, governmental, and private sectors. Partners must be committed to the notion that standards enable innovation, and that open standards enable partnership. This approach is being adopted by more and more businesses (e.g., Google and Amazon through their published APIs). This contrasts with the concept of open source software, which can disenfranchise the private sector and marginalize the enterprise information systems already in place at many government agencies. The goal is neither to force partners to adopt open source software nor to force them to provide open source software instead of a commercial product. Rather, the goal is to achieve interoperability among heterogeneous systems through the adoption of open standards for information exchange that can be implemented with either open source or proprietary solutions.

As the system matures, operations and management needs will change. The number of universities involved in the initial stages of deployment is growing to accommodate interest by other scientists and institutions. The objective is to nurture broad participation in a range of focused activities that leverage and/or contribute to the underlying infrastructure and can become interoperable with other systems as they come on line. Thus, operations and management will involve processes that encourage community participation in both use and evolution of the system.

The SCOOP architecture supports a "virtual organization" comprised of partners that collectively make up a distributed national laboratory for research and applications. In 2005, the SCOOP SOA enabled visualization of ensemble predictions of wave height in New Orleans and Mississippi at www.openioos.org in advance of Hurricane Katrina making landfall (Fig. 4). By the end of 2006, the SCOOP SOA is

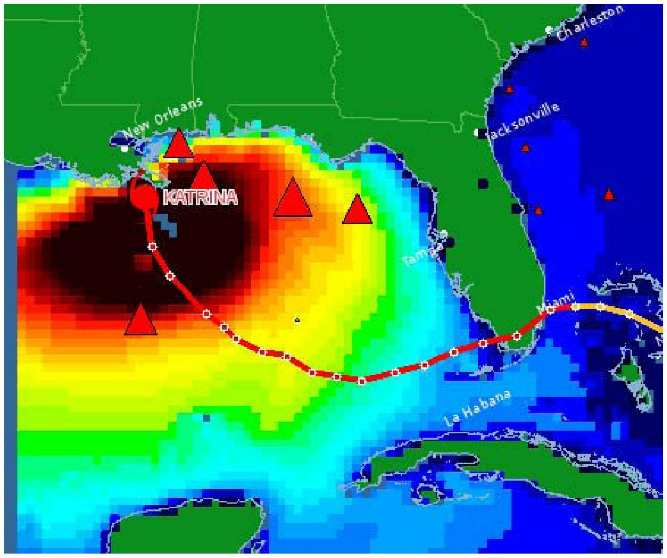

Figure 4: Wave height prediction for Hurricane Katrina on August 29, 2005 as visualized on www.openioos.org

expected to support three additional scenarios: 1) real-time 24/7/365 wave and surge predictions; 2) an atmospheric-model ensemble prediction; and 3) local, high-resolution inundation. Further system refinements are in development. These refinements will continue to advance the accuracy and timeliness of operational forecasts for tidal and storm surge water levels along the Eastern U.S. and Gulf coasts.

## ACKNOWLEDGMENTS

This work is part of the Southeastern Universities Research Association Coastal Ocean Observing and Prediction (SCOOP) Program funded by the Office of Naval Research, Award N00014-04-1-0721 and by NOAA Ocean Service Award NA04NOS4730254. The SCOOP program is grateful to the SCOOP Implementation Team responsible for its development and operation: Bash Toulany, and Yongcun Hu (BIO); Eric Bridger (GoMOOS); Bret Estrada, Chirag Dekate (LSU, Center for Computation and Technology); Steve Thorpe (MCNC); Donna Cote and Matt Howard (TAMU); Ken Keiser, Matt Smith, and Marilyn Drewry (UAH); Justin Davis, Renato Figueiredo, and Vladimir Paramygin (UFL); Lavanya Ramakrishnan and Howard Lander (RENCI); Jian Shen and David Forrest (VIMS); Neil Williams and Geoff Samuels (UMiami); Charlton Purvis; and Mary Fran Yafchak, Don Riley, Don Wright and Joanne Bintz (SURA).