A Management Concept for Ocean Observatories Based on Web Services

A.D. Chave[1], B. St. Arnaud[2], M. Abbott[3], J.R. Delaney[4], R. Johnson[4], E. Lazowska[4], A.R. Maffei[1], J.A. Orcutt[5], and L. Smarr[5]

1) Woods Hole Oceanographic Institution, Woods Hole, MA 02543, USA; achave@whoi.edu
2) Canarie Inc, Ottawa, Ontario, K1P 5M9, Canada
3) Oregon State University, Corvallis, OR 97331, USA
4) University of Washington, Seattle, WA 98195, USA
5) University of California at San Diego, La Jolla, CA 92093, USA

Abstract **- The proliferation of ocean observatories without internationally agreed-upon standards for instrument/user interfaces and observatory control functions is inhibiting interoperability and utiilization of disparate datasets. This is in turn limiting the scientific impact of ocean observatories and increasing their operating costs. Hardware-based standards are not feasible given the different internal architectures of existing and planned ocean observatories. This paper proposes that instrument, data, and observatory control processes be wrapped with web services which will provide a global software standard for these observatory functions. In addition to facilitating interoperability, state-full web services with workflow bindings for observatory instrument and data services will enable dynamic user control of the observatory configuration and the creation of multiple virtual instrument networks within one or more ocean observatories. These concepts are defined and illustrated through a number of use scenarios.**

I. Introduction

Since the 1800's, oceanographers have mapped and sampled across two-thirds of Earth primarily using ships as observational platforms. This has yielded snapshot views of the oceans which have limited resolution in time. Measurements and models from this exploratory phase of oceanography have yielded growing recognition of the diversity and complexity of, and the linkages between, processes that operate above, within and beneath the oceans. The questions posed by oceanographers increasingly cannot be answered using only the tools of the present, in large part because of limited resolution of temporal variability. As a result, the ocean sciences are entering a new phase in which scientists will adaptively observe the earth-ocean system and conduct in situ perturbation experiments. Routine, long term access to episodic oceanic processes is crucial to continued growth in the understanding and prognostic modeling of complex natural phenomena that span an enormous range in space and time.

The evolution of communication technologies coupled with the decreasing cost and size, and increasing capability, of diverse sensors is driving a major paradigm shift in environmental science and engineering. Spatially extended, intelligent, interacting sensor networks are new and appropriate tools for studying complex real-world systems. In the ocean sciences, this paradigm change is being implemented around innovative facilities called ocean observatories which provide unprecedented levels of power and communication to access and manipulate real-time sensor networks deployed within the ocean basins. The real time information flow and data repositories associated with these facilities will empower entirely new approaches to science.

The crucial role ocean observatories will play in 21st century oceanography has received international acknowledgement, and early programs are underway around the world [1-9]. Major ocean observatory infrastructure programs exist in Canada (NEPTUNE Canada), and are under consideration in Japan (Advanced Real-Time Earth monitoring Network in the Area or ARENA), Europe (The European Seafloor Observatory Network or ESONET), and the United States (Ocean Research Interactive Observatory Networks or ORION).

Ocean observatories can be implemented around either submarine telecommunications cables to deliver large amounts of power and bandwidth to the seafloor or with moored buoys to link more limited surface power and satellite bandwidth to the ocean bottom. In either case, the major functional ocean subsystems required to provide instrument services, infrastructure power and communications, and observatory management, including data archiving services, are similar. Existing ocean observatory designs are typically unique, and no generally accepted standards exist for any of these functions. As ocean observatories proliferate and scientists make increasing use of data from them, the absence of standards will limit their utility and hinder broad use of associated data archives.

The purpose of this paper is to outline a unified solution to these issues using web or Grid services distributed throughout the subsystems comprising an ocean observatory, as proposed in [10]. Section 2 outlines the physical subsystems of a generic ocean observatory and describes the services required to link them into a coherent whole. This is illustrated through use scenarios. Section 3 briefly defines web and Grid services and the architecture used to implement them. Section 4 describes a distributed web services architecture for ocean observatories. The flexibility of the resulting network recursive architecture is outlined with use scenarios and by contrasting it with a conventional stovepipe

0-7803-8669-8/04/$20.00 ©2004 IEEE.

architecture. Section 5 reviews some outstanding issues.

II. Components of an Ocean Observatory

Reference [11] gives a comprehensive overview of the design of cabled ocean observatories, and only a summary will be given here. Fig. 1 is a block diagram showing the key hardware subsystems of an ocean observatory science node. Copies of these subsystems appear both in other science nodes and in the shore station, either of which is a logical peer. While the figure is directly pertinent to a cabled observatory, the same subsystems will appear in a moored installation except that the submarine cable to other nodes or a shore station is replaced by a buoy-to-satellite communications link and an autonomous power source in the buoy.

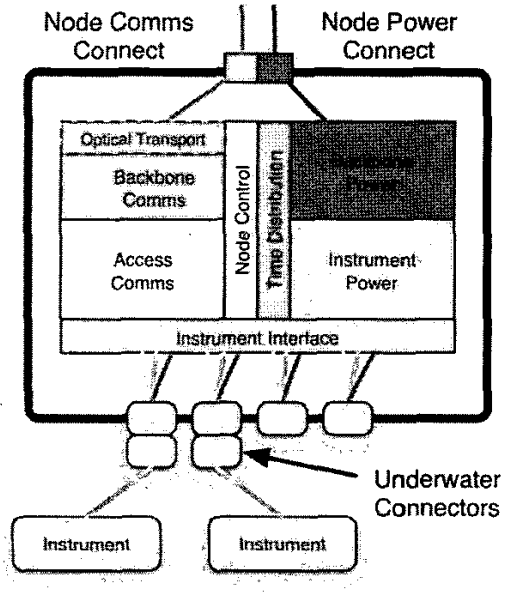


Fig. 1. Block diagram showing the main subsystems in a cabled ocean observatory seafloor science node. See text for discussion.

The major components of an ocean observatory science node are power, data communications, node control, time distribution, and instrument interface subsystems. The power subsystem transforms high voltage DC carried by the single conductor (with seawater return) present in standard submarine fiber optic cable to lower levels usable by both the node infrastructure and science users. This usually will be handled in two stages, as shown in Fig. 1. The data communications subsystem may consist of an access (e.g., layer 2 Ethernet) switch which aggregates IP streams from a diverse set of instruments and transfers them to a backbone (operating at layer 2 or 3, depending on the observatory topology) router. The backbone router also handles data streams arriving from or going to adjacent science nodes. The optical transport in the backbone communications subsystem interfaces to the backbone fiber optic cable. Both the power and data communications systems are typically redundant in the interest of reliability.

The node control subsystem (NCS; Fig. 1) provides high reliability oversight of all node functions, including telemetry of critical data and master control of key subsystems. The NCS may be distributed among the other node subsystems or centralized, depending on the implementation. The time distribution system serves synoptic, high accuracy (e.g., 1 microsecond or better) time corrected for latency to science users. Finally, the instrument interface subsystem (IIS) provides power, communications, and time services to instruments and sensors, and may be physically distributed between the science node and individual instruments. The instrument interface contains part of the seafloor component of the data management subsystem (DMS) which is logically distributed throughout the observatory. The remainder of DMS is contained in the shore station or other terrestrial sites and in the NCS.

Fig. 2 is a functional block diagram showing the key processes required to implement an ocean observatory that comprise the observatory management system (OMS), which encompasses the NCS, IIS, and DMS. This figure emphasizes interfaces, and the required processes will typically be implemented in software with associated hardware elements.

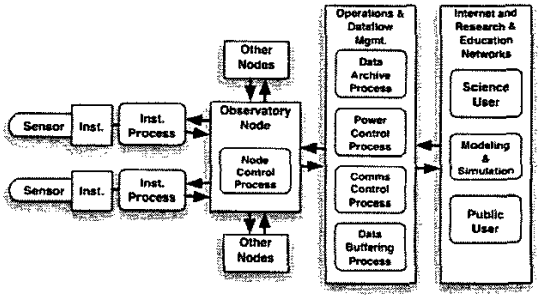


Fig. 2. Block diagram defining the distributed processes required to implement a comprehensive Observatory Management System. See text for discussion.

An instrument process resides logically between the instrument and the access data and user power connections at the node. Some of its functions include: monitoring and control of power to the instrument; instrument ground fault detection; bi-directional transmission of data to/from other instruments, other nodes, and the shore station; service of instrument metadata to users or data repositories; and acquisition and processing of synoptic time from the infrastructure for use in the instrument. The instrument process may physically reside in the instrument itself, in the node between the wet-mateable user port and the node data/power hardware, or even in proxy mode on shore.

The node control process may interact with many

instrument and shore-based processes. A partial list of the node control process functions includes: monitoring and control of the node power busses; monitoring and control of the node access and backbone layer data communications hardware; and collection and transmission of node engineering data to shore. The node control process may reside physically in the node or on shore.

An ocean observatory operations center on shore houses all of the servicees to monitor, maintain, control, and manage the myriad components of an ocean observatory. The operations center connects to a set of distributed data archives which are probably not co-located. There are four main types of shore-side processes: the communications control process, the power control process, the data buffering process, and the data archive process.

The communications and power control processes are used to control numerous node, instrument, and sensor functions, and may interact with both the node control and instrument processes. A partial list of their functions includes: monitoring and control of system-wide power usage and availability; monitoring and management of system-wide data bandwidth usage and availability; and creation and elimination of connections between instruments or nodes and other processes.

The data buffering process gathers real-time data from instruments, as well as engineering data from the nodes, and provides short term storage in the event that connectivity to the Internet and distributed data repositories is interrupted for any reason. In ordinary operation, it simply serves a pass-through function.

The data archive process receives data and/or metadata from instruments via the data buffering process. Different repositories may receive and process data from different types of instruments. A partial list of the data archive process functions includes: extraction of instrument metadata from the instruments; acquisition and post-processing of data streams from instrument processes; and possibly provision for national security control over data access.

The different classes of processes interact in complex ways which may be understood through use scenarios. The following illustrate some of the issues.

Scenario #1: Assume that an instrument at node X on a multi-node ocean observatory needs to turn on a light. The associated instrument process cannot be allowed to do this directly because if insufficient power is available at node X, then other instruments or the entire observatory could be affected. The instrument process queries the node control process in node X. The node control process checks the request against the available power, rejecting it if the reserve is known to be insufficient and approving it otherwise. The node control process is updated regularly by the shore-based power control process which maintains a system-wide model of power availability. This is necessary because the power system comprises a complex set of distributed voltage drops along long runs of submarine cable, and hence local changes in power consumption can have wide-ranging effects. Thus, the simple act of turning on a light requires interaction of three types of services distributed widely around the observatory.

Scenario #2: Assume that a data assimilation model running at a user's institution is interacting with a subset of the instruments at one or more nodes on one or more ocean observatories. The model is capable of altering the data collection protocols, and hence has to interact directly with the associated instrument processes. It also has to interact with the data archive processes on shore. For example, if the modeling program alters instrument parameters, then metadata will be updated in the data archive, and this information will have to be inserted into the model. Deeper levels of interaction can also be envisioned; changes in instrument demands for power or bandwidth may involve the communications and power control processes, as well as many node control processes.

These use scenarios outline the complexity of the control and interface functionality needed to operate an ocean observatory. The complexity increases with the size of the installation, and is certainly greater for a multi-node observatory than for a single cabled node because of the need to coordinate processes across the entire facility. While many of the observatory processes could be implemented in hardware, a common characteristic is the need for repetitive, automatic inter-process communication and the consequent exchange of data or commands. In addition, it is certain that the required processes will evolve over time as unforeseen modes of operation emerge. This argues strongly for an approach which places hardware interfaces at primitive levels in the subsystems and implements the inter-process links and their control in software which can be modified as required. Finally, as ocean observatories proliferate, the need for interoperability will become increasingly important. This is especially true at the sensor and user interfaces where standardization will be essential to wide use of instruments and data. However, interoperability at the internal levels of ocean observatories will lead to a reduction in operating costs as the community-wide experience base grows.

As an illustration of a comparatively inflexible ocean observatory design, Fig. 3 depicts the data system for a "stovepipe" observatory architecture which was typical of large autonomous data collection science systems in the recent past. A common characteristic is that once the design is completed, it is fixed for the lifetime of the project without a major re-design and re-build which typically involves basic architectural changes. In Fig. 3, the data path consists of a single line from a seafloor instrument, through the node communications subsystem, to the shore station and onto the Internet, terminating at a shore-based data archive. Services are largely fixed and unchangeable throughout the wet and dry parts of the system, and hence there is limited flexibility in

managing the observatory. Recursive modes of operation such as in the second use scenario above cannot be implemented with this architecture because the required processes cannot be linked dynamically. In addition, distributed observatory processes cannot easily be added to such an architecture after construction without extensive retrofitting of the wet plant. Instead, a comprehensive strategy for ocean observatory interface and control processes has to be devised early in the design phase.

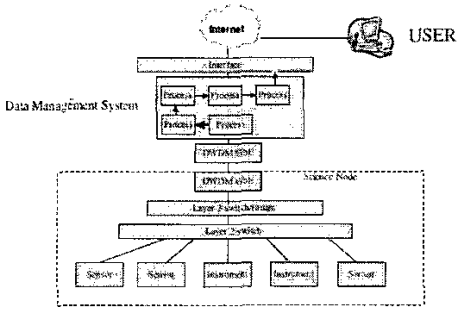


Fig. 3. Architectural block diagram showing a traditional stovepipe design for an ocean observatory, in which instrument, control, and data management and archiving processes are hardwired with fixed interfaces that cannot be altered.

III. Web and Grid Services

Web services is an emerging distributed computing paradigm which integrates the description of software components with a means to access them and discovery methods to find them using simple Internet standards. Web services function across hardware platforms, operating systems, or programming languages. Web services extend existing web protocols, which are designed to interface computer-based text and graphics to a human user through a web browser, to direct, computer-to-computer communication and software execution. Web services can operate in either a message-oriented or a procedure-oriented mode. For an overview, see [12]. Web services specifications are being defined within standards bodies and are spawning new industries.

Web services is implemented using the Extensible Markup Language (XML)-based Web Services Description Language (WSDL) which provides a start-to-finish framework that operates on messages containing either documents or procedure calls. WSDL defines the abstract transformations necessary to send and receive data or commands to underlying, wrapped software, and in turn binds them to network protocols to define the endpoints. WSDL contains three main parts. The data definitions component specifies the structure of the data. The transformation component determines any operations to be performed. Finally, the service bindings component specifies the network transport and ports that will be used to transfer the result to a destination computer.

The Simple Object Access Protocol (SOAP) is the emerging dominant messaging framework for web services communication, enabling the one-way transmission of a message from source to destination computer. The SOAP standard specifies a minimum level of complexity required to send a message, but is extensible to more elaborate applications and protocols. SOAP is mappable onto standard IP transports like HTTP or FTP.

The final component of the web services architecture is a web services registry and discovery system which publishes and serves pointers to web services interfaces. This is facilitated through Web Services Inspection Language (WSIL) documents which contain service descriptions (i.e., URLs to WSDL documents) and links to other archives of service description. Service descriptions may also be entries into Universal Description, Discovery, and Integration (UDDI) catalogs of web services, although this business-oriented discovery mechanism is often inadequate in a research environment. A more comprehensive Web Service Discovery Architecture (WSDA) is described in [13].

The initial early application for web services was linking legacy transaction business applications and document processing with the Internet. Transaction interaction implies that no state is maintained after a process is completed. For example, when a web service system processes a purchase order, all of the information that is needed to understand the purchase order is included as part of the WSDL message describing the transaction. However, researchers recognized that web services could be useful in state-full applications. This was first identified by the global Grid community, who wanted to use web services to enable sharing of high performance computers among a distributed user base. They defined Grid services, which are an extension of web services that includes management of state, identification, session, and lifetime, along with a notification capability. When sharing a computer, or in other long time duration applications, state must be maintained until use of the resource or application is completed. This requirement led the Global Grid Forum to develop the Open Grid Services Architecture (OGSA; see http://www.ogsa.org/) which provided for state-full web or Grid services [14-15]. This class of web services has now been fully encompassed by the new Web Services Resource Framework (WSRF) standard. The addition of state-full web services allows the web services paradigm to be applied to a whole set of new applications. State-full web services allow the web services paradigm to be applied to instruments and other physical resources in addition to software processes; analogous efforts in related fields include the Grid Middleware Instrument project at Indiana University

(http://www.instrument-middleware.org) and the European GridCC project (http://www.lnl.infn.it/~maron/gridcc/Annex_I/gridcc_annexI_vs_1.5_Approved_V2.doc).

The final and most recent development in web services is workflow tools that allow users to construct complex web services from a set of simpler ones; in essence, the user orchestrates the dynamic interconnection of diverse web services. The recently introduced Web Services Flow Language (WSFL) and Web Services Orchestration Language (WS-ORCH) will standardize workflow, and is often referred to as a critical part of emerging distributed computing environments. WSFL and WS-ORCH capture the linkages of constituent services together to build larger composite services.

IV. A Web Services Architecture for Ocean Observatories

Ocean observatories are a classic multidisciplinary research infrastructure which encompasses both legacy and new installations ranging from individual moored buoys to multi-node regional cabled observatories. To date, there is very little hardware or software commonality among existing or planned systems. Each has unique external sensor and control protocols, internal power and communications subsystems, and data archive capabilities. Interoperability is limited; instruments typically have to be re-interfaced to move between observatories, and custom software is often required to access data from different facilities. However, it is widely recognized that maximizing the science value of ocean observatories requires commonality so that scientists from diverse disciplines can share instruments, data, and ultimately ideas. This reflects an increasing global trend for large science projects to build an overarching cyberinfrastructure that makes data collection and analysis feasible for a broad user community.

In [10] and this paper, a novel approach is proposed in which state-less and state-full web services and workflow tools are used to create a cyberinfrastructure facility for ocean observatories. This vision encompasses the main elements of data management and archive, observatory management, and instrument interface processes, and substantially transcends any one of these taken alone. As a consequence, rather than having each ocean observatory construct a single central data archive to collect all sensor and instrument data, as depicted in the stovepipe architecture of Figure 3, multiple distributed data processes would be created using web services as the common interface technology. Perhaps more significantly, this architecture enables and unifies DMS-NCS-IIS interactions in a flexible and extensible manner.

In the proposed architecture, instrument web services may be exposed as a single service, or they may be a collection of services aligned with disciplinary groupings of instruments, or a collection of services aligned with a multidisciplinary experiment. Multiple web services can be associated with single instruments, and hence multiple virtual observatories within an ocean observatory can be established or removed dynamically. Each independent web service is made up of one or more science instruments, network connections, data management systems, and a distributed observatory management system. Aggregations of web services can also be re-advertised as new composite web services.

For legacy instruments not explicitly designed for the proposed style of observatory operation, proxy web services can substitute for embedded web services. With proxy web services, the actual interface is hosted at a shore station, but to the science user, communication appears to be directly with the instrument. Alternately, a standard science instrument interface that goes between a legacy sensor and the node instrument interface which incorporates embedded web services could be provided by the observatory operator, analogous to the puck proposed in [16]. For either embedded or proxy users, the bindings and interfaces between the various ocean observatory components are not specified at the design stage, but are left to be defined by the scientific user and observatory management team at the operational stage using work flow tools. The proxy model can also be implemented on shore to minimize bandwidth utilization and power requirements where required (such as in buoyed installations).

In the proposed architecture, all science instruments, data management services, and some observatory management services will be implemented as independent, unique, state-full web services. In most instances, observatory management services will be state-less; an instrument or a shore control process will request some operation and disconnect once it is completed. The science user binds the different state-full processes together at operation time using web services work flow tools. State-less control connections are established and removed automatically as needed. This provides the greatest architectural flexibility and easily allows changes to be made to incorporate new experiments without affecting the data flow from existing experiments. In essence, the web workflow approach allows users to bind processes together as required for particular applications largely independent of (but perhaps limited in the end by) the original design and implementation.

A state-full web services compliant scientific instrument and data management service (whether as a proxy or embedded) would have WSDL descriptors and service handle implementation for a number of service ports including:

(a) Ocean observatory management, monitoring and data archiving

(b) National security authority monitoring and management

(c) User management and monitoring, which may support multiple ports for multiple users
(d) User data flow
(e) Calibration data flow

Ocean observatory state-full web services could include network services (IP, port, broadcast, multicast, virtual private network (VPN) or lightpath), data logging services, calibration logging services, data archiving services, metadata cataloging services, and many others.

When a science user wishes to set up an experiment, they will use a standard web services workflow tool to bind together a number of instruments and services to create a complete data system for that specific experiment. For example if a science user needed a live video feed from a specific camera, they would discover the appropriate web services-compliant video instrument service, bind that to an advertised ocean observatory network service (e.g., a VPN with appropriate quality of service), in turn bind that to a shore station data archiving service to capture and store the raw data feed, then bind that to a lambda-switched lightpath service across a dedicated terrestrial scientific network, then bind that to an MPEG 7 encoding service, and finally bind that service to a science user's own web services-compliant data assimilation model somewhere else in the world.

Fig. 4 illustrates the web services approach. Every network element, sensor, instrument and control process is exposed and registered as web services (either state-full or state-less, depending on their nature). There is no attempt to pre-define the linkages between the various elements. Instead, the science users, their instruments or models, and the observatory operator through the shore control processes, have the freedom and flexibility to link or bind these various services together either automatically or using work flow tools. Different science users can simultaneously bind disparate services together in different ways.

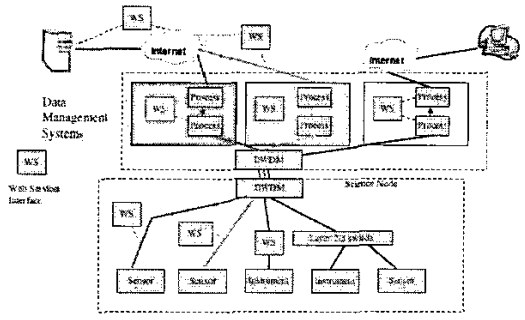


Fig. 4. Architectural block diagram illustrating the components of a flexible ocean observatory design utilizing a web services architecture with workflow bindings.

For example, in Fig. 4, the red science user can bind a specific instrument with a specific local area network service which in turn is bound to a specific network service which is finally bound to a set of Internet services to deliver the data to a remote high performance computer. Simultaneously, the green user can utilize the same technologies to create a different set of linkages using the same or different instruments, network services and software processes. The black user can simultaneously use other instruments and observatory services to carry out a different experiment.

Web services also serve as the integrating technology to seamlessly connect disparate ocean observatories together into coherent virtual ocean observatories (VOOs). To carry this out, individual ocean observatory operators need to wrap existing proprietary instrument, control, and data handling processes with standard web services and advertise the results. A user can then bind instruments at multiple ocean observatories into a VOO. A simple use scenario where this would be scientifically enabling is the transit of a major storm system up the eastern seaboard of North America, during which it would cross a series of coastal ocean observatories. By binding these observatories together into a VOO, the growth and decay phases of the ocean's response to an extreme atmospheric forcing event could be continuously studied and/or used for predictive purposes. The concept could readily be extended to provide a web services wrapper for CODAR or weather radar so that these data types could be incorporated into an experiment.

V. Discussion

A prototype example of the architecture described for ocean observatories is the CA*net 4 User Controlled LightPath (UCLP) [17-19], which provides user switching of dedicated wavelengths on a long haul research network. With UCLP, individual research users can link together various network elements in any way they want to create their own private network topology. The ocean observatory architecture is an extension of this concept to cover a more complete set of services where not only network elements, but also instruments, sensors, and control processes, can be linked together into complete systems.

One of the biggest challenges for projects like ocean observatories is the great degree of uncertainty in the design requirements for the access and backbone networks and the power system to provide services that must be useful for decades. It is very difficult to predict what new services and facilities will be needed or what instrumentation will be deployed in coming years. The use of state-full web services and work flow tools for observatory functions that users interact directly with, and state-less web services for background services, allows researchers and operators to think of the ocean observatory infrastructure as a set of simple building blocks. In the past, hardware and software engineers would spend years designing a detailed architecture

to support this type of scientific research, as is implicit to the architecture in Fig. 3. With web services and work flow tools, the infrastructure can be reconfigured at any time to meet the needs of future researchers, or to incorporate lessons learned. This argues strongly for the simplest possible, modular design for the access and backbone networks to maximize their adaptability to future needs.

In principle, extending the web services concept to a flexible facility like an ocean observatory is straightforward. The complexity will arise in meshing diverse state-full and state-less web services into a coherent whole on a scale that has not previously been attempted, but even this task is simpler than building a reliable, fixed system such as that in Figure 3 which will serve science needs for long periods of time. A key distinction is recognition that, for the first time, the architecture of a research infrastructure will be dynamic and in the hands of the science users for the life of the project rather than being locked in place years before it starts by software and hardware engineers who cannot be expected to foresee science applications decades into the future.

REFERENCES

[1] Momma, H., R. Iwase, K. Mitsuzawa, Y. Kaiho, and Y. Fujiwara, Preliminary results of a three-year continuous observation by a deep seafloor observatory in Sagami Bay, central Japan , *Phys. E. Pl. Int.*, vol. 108, pp. 263-274, 1998.

[2] Delaney, J.R., G.R. Heath, B.M. Howe, A.D. Chave, and H. Kirkham, NEPTUNE: Real-time ocean and earth sciences at the scale of a tectonic plate, *Oceanogr.*, 13, 71-83, 2000.

[3] Favali, P., et al., Towards a permanent deep sea observatory: the GEOSTAR European experiment , in *Science-Technology Synergy for Research in the Marine Environment: Challenges for the XXI Century*, L. Beranzoli, P. Favali, and G. Smriglio, Eds. Amsterdam: Elsevier, 2002, pp. 111-120.

[4] Schofield, O., et al., The Long-term Ecosystem Observatory: An integrated coastal observatory , *IEEE J. Ocean Eng.*, vol. 27, pp. 146-154, 2002.

[5] Hirata, K., et al., Real-time geophysical measurements on the deep seafloor using submarine cable in the southern Kurile subduction zone , *IEEE J. Ocean Eng.*, vol. 27, pp. 170-181, 2002.

[6] Austin, T.C., et al., A network-based telemetry architecture developed for the Martha's Vineyard Coastal Observatory , *IEEE J. Ocean. Eng.*, vol. 27, pp. 228-234, 2002.

[7] Petitt, R.A., Jr., et al., The Hawaii-2 Observatory , *IEEE J. Ocean Eng.*, vol. 27, pp. 245-253, 2002.

[8] Shirasaki, Y. et. al., "ARENA: A Versatile and Multidisciplinary Scientific Submarine Cable Network of Next Generation", *Proc. 3rd Int. Workshop on Scientific Use of Submarine Cables and Related Technologies*, Piscataway: IEEE, pp. 226-231, 2003.

[9] M.Yoshida, and Y. Hirayama, "The data transmission system for the real-time seafloor monitoring cable network", *Proc. 3rd Int. Workshop on Scientific Use of Submarine Cables and Related Technologies*, Piscataway: IEEE, pp.197-200, 2003.

[10] St. Arnaud, W., et al., "An integrated approach to ocean observatory data acquisition/management and infrastructure control using web services", *Mar. Tech. Soc. J.*, in press, 2004.

[11] Chave, A.D., G. Waterworth, A.R. Maffei, and G. Massion, "Cabled ocean observatory systems" *Mar. Tech. Soc. J.*, in press, 2004.

[12] Newcomer, E., *Understanding Web Services*, Boston: Addison-Wesley, 2002, 332 pp.

[13] Hoschek, W., Peer-to-peer Grid databases for Web service discovery , in F. Berman, G.C. Fox, and A.J.G. Hey, Eds., *Grid Computing: Making the Global Infrastructure a Reality*, Chichester: John Wiley, 2003, pp. 491-542.

[14] Foster, I., C. Kesselman, and S. Tuecke, The anatomy of the Grid , in F. Berman, G.C. Fox, and A.J.G. Hey, Eds., *Grid Computing: Making the Global Infrastructure a Reality*, Chichester: John Wiley, 2003, pp. 171-198.

[15] Foster, I., C. Kesselman, J.M. Nick, and S. Tuecke, The physiology of the Grid , in F. Berman, G.C. Fox, and A.J.G. Hey, Eds., *Grid Computing: Making the Global Infrastructure a Reality*, Chichester: John Wiley, 2003, pp. 217-250.

[16] O'Reilly T., et al., "Smart network" infrastructure for the MBARI Ocean Observing System , Proc. IEEE Oceans 2001, Piscataway: IEEE, pp. 1276-1282, 2001.

[17] St. Arnaud, Bill, J. Wu, and B. Kalali, Customer-controlled and managed optical networks , *J. Lightwave Tech.*, vol. 21, pp. 2791-2803, 2003.

[18] Boutaba, R., W. Golab, Y. Iraqi, and B. St-Arnaud, Lightpaths on demand: A web services based management system and case study , *IEEE Comm. Mag.*, vol. 24, pp. 101-107, 2004.

[19] Boutaba, R., W. Golab, Y. Iraqi, T. Li, and B. St. Arnaud, Grid-Controlled Lightpaths for High Performance Grid Applications , *J. Grid Comp.*, in press, 2004.

© Copyright 2004 MTS/IEEE