USING XML TECHNOLOGY FOR DATA AND SYSTEM METADATA FOR THE MBARI OCEAN OBSERVING SYSTEM

Daniel Davis, Tom O'Reilly, Duane R. Edgington, Rich Schramm, and Karen A. Salamy
Monterey Bay Aquarium Research Institute
7700 Sandholdt Rd. Moss Landing, CA, 95039 USA
E-mail: dada@mbari.org

Abbreviated Title: Using XML for the MBARI OOS
Keywords: ocean observing systems, XML, metadata, smart networks, UUV.

Abstract
The Monterey Bay Aquarium Research Institute (MBARI.ORG-WWW*), a nonprofit, privately funded, research institute devoted to the development of oceanographic technology has been developing systems for long term environmental monitoring in Monterey Bay since 1987. The institute has initiated a project for expanding its ocean observing system capabilities through an expansion of existing moored data acquisition systems. The goal of the MBARI Ocean Observing System (MOOS) project is to develop highly flexible, configurable, and redeployable, ocean observing systems utilizing a wide variety of sensors, instruments and platforms, including moorings, ocean-bottom substations, and unmanned underwater vehicles (UUVs), to provide semi-continuous observations of important physical, biological, and chemical variables extended in space and time to support long term monitoring and event detection, such as the onset of an El Nino, as well as support for focused, intermediate-term, scientific process studies.

A primary challenge of this effort is the development of a software architecture to manage, control and access a variety of sensors, instruments, and platforms and manage the movement of data over a wide variety of communication links to users. The subsystem that performs these functions is called the MOOS Instrument Software Infrastructure (ISI). An associated challenge is to insure that all the ancillary data, the metadata, required to interpret both system and scientific data produced by the system are also properly defined, managed and made accessible to users along with the data streams. The approach taken by MBARI is to utilize XML technology to address this problem.

We provide an overview of the MOOS ISI system design and an overview of the scientific and system requirements that must be met by the metadata architecture. We describe a simple prototype process developed at MBARI and based on XML that can be used not only to define classes of metadata, but also to design user-friendly metadata forms to create, revise and access XML based metadata documents to satisfy a variety of requirements for the observing system.

1.0 Introduction

There is a steadily increasing interest in the use of ocean observing systems for monitoring the health and conditions of the worlds' oceans. Both international and national efforts to design and implement such systems are well underway (Malone and Cole (2000); Nowlin et al, (1996); GOOS-WWW*; OCEAN.US-WWW*). A variety of new technologies hold promise for resolving some of the difficult technical issues and challenge these types of systems pose. First and foremost is the use of web technology to provide broad user access to the observational products of these systems. Less known, but also important, is the advent of 'smart networks', which incorporate the automatic ability to reconfigure an observing system with new platforms and sensors through the concept of plug and work'. All these technologies are based on modern notions of object-oriented software design and development for distributed systems.

* Full web site URLs are listed in the References

A companion innovation to this revolution in software is new technology for managing information over distributed systems. The technology is an adaptation of general markup languages which have evolved from origins in document processing to extensible markup languages (XML) for managing information over distributed, highly configurable, network based systems (XML.COM-WWW). Just as these other web-based technologies are essential to the design and implementation of ocean observing systems, XML is gaining attention as an enabling technology for managing and accessing the information that observing systems produce (MEDI-WWW; MARINE.XML-WWW). A central feature of this technology is the ability to describe data independently of how the data is ultimately presented. Thus XML has the potential for being a universal language for describing data used over networks irrespective of how the data is ultimately used or displayed.

As ocean observing systems evolve, the types and form of the data they produce will evolve as well. This poses the challenging problem of utilizing and comparing similar or equivalent data over time in the presence of an ongoing evolution in sensor, instrument, and platform technology. A key to solving this problem is the early capture of metadata, along with the data, in a form that enables the scientific interpretation of the data over time.

Therefore a primary challenge of the metadata problem is providing highly flexible, easy to use solutions that can respond readily to changes in systems and the data they produce and are yet sufficiently simple that they have the potential of being widely used and disseminated. Describing specific metadata solutions for existing types of data is not the primary focus of this paper. The focus here is on a process that exploits existing XML technology to provide simple, easy to use methods to define, create and use metadata and to illustrate a prototype of this methodology for a system currently in development and planned for deployments in the immediate future.

2.0 Overview of the MBARI Ocean Observing System

A system diagram for MOOS is shown in Fig. 1. The system now under development will enable

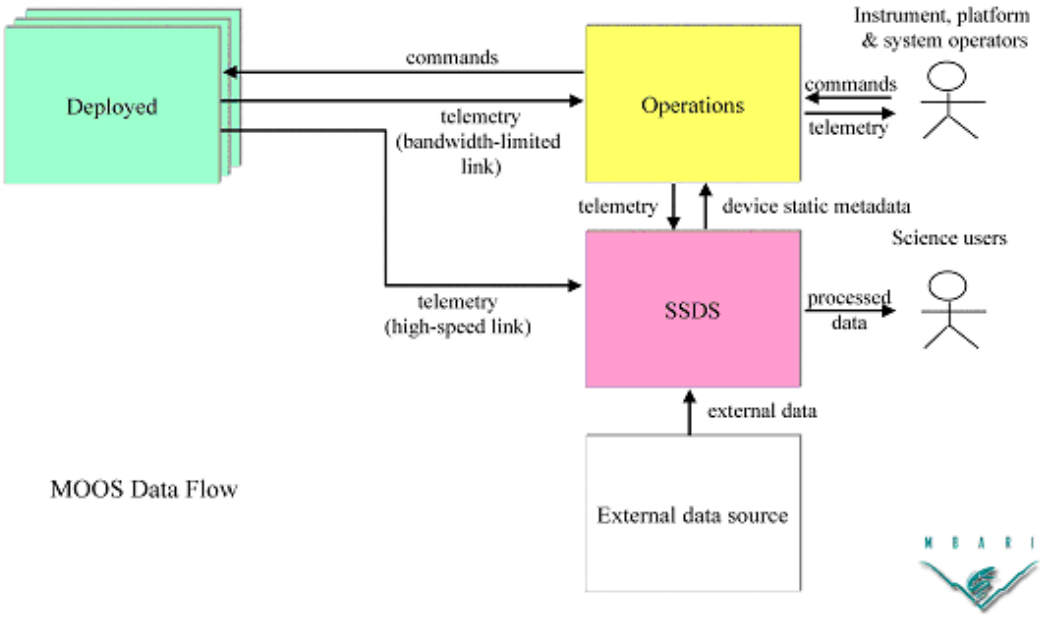

Fig 1. System block diagram for MOOS

coordinated data acquisition from a diverse set of sensors, instruments and platforms. MOOS network software is already in the early stages of implementation at MBARI. The system requirements that are driving the design and implementation are the result of several years of analysis of the proposed scientific uses of the system, and discussion among scientists as well as ongoing deployments of prototypes dedicated to specific science experiments (MBARI.ORG-WWW). The required capabilities of MOOS present some very interesting engineering challenges. In this paper, we will look at these challenges from the software perspective, and review technologies both available and proposed, which may provide solutions. It is not the purpose of this paper to completely review the requirements and goals of the MOOS system but only to summarize the elements of the system required to provide a context for the subject of this paper.

A portable mooring system, with attached seafloor fiber optic network is the first element of MOOS; this platform will host surface, midwater, and benthic instruments, as well as a docking station for autonomous underwater vehicles (AUVs). The mooring will be but one component of a much broader ocean observing network, which incorporates ships, ROVs, AUVs, drifters, and other instrument platforms. These platforms (some already existing, some yet to be developed) will enable data collection over a broad geographic area and throughout the oceanic water column, from the sea surface mixed layer and euphotic zone, through the midwater, to the deep ocean seafloor. The instruments themselves will range from current-off-the-shelf (COTS) instruments such as CTDs and fluorometers, to novel custom-made devices developed at MBARI and elsewhere. Instrument command interfaces are very diverse, as currently there are no widely accepted interface standards; thus the MOOS network software architecture must accommodate a variety of protocols.

2.1 The Deployed Subsystem

The deployed subsystem is the part of the system deployed at sea and can consist of moorings, ships, UUVs and a benthic network of substations connected to the surface by fiber optic links through a riser to a mooring, and then by radio links, or direct cable links, to shore. The subsystem nodes may also include a variety of UUVs in intermittent acoustic communication with local subsystem elements. UUVs may also 'dock' with subsystem elements and directly download data for transmission to shore through communication links. The deployed subsystem is also referred to as the 'wet side' of the system.

2.2 Shore Side Data Subsystem

A companion to the MOOS 'wet side' is a shore side data system (SSDS); also referred to as the 'dry side'. The SSDS is to be the repository for all the scientific data collected, and to provide users with access to this data. A primary requirement is to insure that all the metadata required for properly describing and interpreting the data from the system instruments is collected, catalogued, and accessible along with the data. In the context of a highly flexible and reconfigurable observing system such as MOOS, this problem is particularly challenging. The SSDS also provides a capability for accessing data sources external to the MOOS system.

2.3 Operational Subsystem

A third aspect of the MOOS system is the operational need to monitor, control, diagnose and recover from system failures in 'real time'. This places further demands on the MOOS architecture. The architecture must be capable of capturing the current status of system elements (sensors, instruments, platforms and communication links), as well as controlling and modifying them. An audit trail or 'history' of the system state is also required to support diagnostics and recovery and to fully identify the system state as an aspect of data interpretation. Additional system metadata will need to be defined and accessible to support the operational subsystem.

2.4 Smart Network Technology and the Instrument Software Infrastructure

A fundamental feature of the MOOS effort is the utilization of recent advances in 'smart network' technology to design an instrument software infrastructure which provides real-time reconfiguration, remote device control, and automated event detection and response within a network with limited bandwidth links (radio-frequency and acoustic). Smart networks also provide a capability for 'plug and work' instruments, or automated device and service discovery, based on a distributed object-oriented software architecture. The flexibility for reconfigurability of the system created by the use of this technology must also be reflected in the approach to metadata issues. The software infrastructure being designed to support these features is called the MOOS Instrument Software Infrastructure or MOOS-ISI (O'Reilly *et al*. (2001)).

3.0 The Metadata Problem

The conventional view of metadata is that it is ancillary information required to properly interpret a data set or data stream. It is generally discussed with reference to *scientific data*. Sometimes it is used in reference to catalogs of data used to search for and access datasets. An example is the Marine Environmental Data Inventory system developed and maintained by IOC (MEDI-WWW). Here we consider metadata only in an ocean observing system context, which not only includes metadata for the scientific data streams, but also metadata for the observing system as a whole. That is, any data required for interpreting the data from the system, or any data required to manage, control, or describe the state of the system at any time, will be viewed as metadata. Some of this 'system metadata' may be useful scientifically as well.

Any methodology used to address the metadata problem must provide methods to define, capture, access, and represent the metadata. It is commonly understood that the forms that metadata can take vary greatly. The fact that a certain person or laboratory performed the calibrations of an instrument may be important information when it comes to interpreting data. This is particularly the case in the presence of new sensor development. A description of the processing of raw sensor data to create data streams may be important metadata. The fact that a specific sensor came from a particular laboratory, or manufacturer at a certain time, may be critical. The fact that other data streams are available along with the data stream of interest may be another critical piece of information. Or even the status of the observing system, such as the results of automated event detection or system failures at critical times of observation, may be important information to properly interpret data. Thus the first requirement of any methodology used to define metadata is that it must be capable of describing information in a wide variety of forms. There is a need to represent a wide variety of data structures: numeric, string and character data, as well as lists, arrays, tables, and hierarchically structured and other structured forms of data.

Another critical challenge is to insure that important metadata is defined and entered as soon as a system element becomes operational. It must be easy to define and create important metadata for the system to support early metadata capture, or to easily modify metadata whenever the system is modified. There are numerous standards efforts dedicated to carefully describing what metadata may be required for various types of scientific data, but unless it is relatively easy to implement such standards, the metadata may not be created at all, or easily accessible to users of the system.

Another requirement is that even though users or systems may capture the same essential information in different forms, it must be relatively straightforward to convert from one form of metadata information to an equivalent, but different, form. And there must be a capability to view and display data and metadata in a wide variety of ways. This means that the methods used to represent and store information may be quite separate from the means by which the information is displayed or viewed. In particular, it should be possible for users to view and interact with the metadata in a user-friendly

manner, that is, a form natural to a particular user. Thus there should be simple interfaces to define what needs to be captured, for capturing it, and for accessing and displaying it after it is captured. Given that ocean observing systems are evolving but may be deployed over long periods of time, or reconfigured for shorter deployments, it is necessary to have a technology that provides for incremental development and for readily modifying, without completely redoing, what has already been done.

Finally it is critical that the technology used to represent metadata does not constrain, imply, or define a particular *science policy on the interpretation of data*, but provides the widest variety of mechanisms for implementing policies on how metadata may be used to interpret data. In short, the technology should not constrain but instead facilitate what is most natural to its users.

4.0 A Prototype Process for using XML for metadata

The primary concept behind using XML for system metadata is that this technology provides a flexible and adaptable language and technology for defining, managing, rendering and displaying the data in a variety of ways. The features of XML technology provide opportunities to address a number of challenging problems in the metadata capture and management problem. There is widespread recognition that the technology meets the capabilities required by the metadata problem as outlined above. And there is already a rich tool-set of applications that can be utilized to address many of the challenging problems. A concern shared by some is that it would be unrealistic for users to learn the specific syntax and theory behind XML in order to obtain its benefits. A key to solving this problem is to have user friendly interfaces created by developers that eliminate any need for users to master the technical aspects of XML technology. In this way users can benefit from the features of XML without necessarily requiring them to know much about its details.

In the case of the MBARI Ocean Observing System we require mechanisms that allow scientific users to create, manage, and access metadata for scientific data. But we also need to provide operational users an equal ability to create information about sensors, instruments, platform, communication links between sensors, instruments, or platforms and shore, and the status of system elements that meet system and operational needs as well.

We propose a prototype process that involves two basic steps. The first step is carried out by designers who define the types of metadata structures for the *classes* of objects that make up the system. Along with each object class, definition designers create user-friendly forms that can be used by the operational and scientific users of the system to create specific instances of these objects for the actual system. The goal of the first step is to specify what kinds of metadata are required by users for a class of sensors, a class of instruments, a class of platforms, or a class of communication links.

In summary the prototype process we have developed at MBARI for using XML in MOOS are:

1) A developer or designer uses an existing application like XML Spy to design a XML schema to describe a type of metadata needed for a particular class of system object, whether it is scientific or operational. The designer must take into account what information the user needs for objects in this class.
2) The developer or designer uses the schema to design a basic *user interface class form* for users to create or access a specific instance of the metadata described by the schema. The methodology illustrated below uses the visual interface design to create a JAVA application that displays a form that can be easily used to generate, view or revise, the XML metadata documents for each system object. Although the metadata may be stored in XML form, or even a compressed form, it is created, viewed and revised only through metadata object class forms.

3) System users, scientists, technicians and system operators use the forms for each class of system object to generate and view specific XML metadata without any requirement for the user to know or understand the XML language itself. Developers or designers can also develop other views, including partial views, of the metadata for specific purposes using the same tools. Designers can also design and develop converters using XSL (XML.COM-WWW) to create graphic displays or transform from one form of data to another.
4) The XML system metadata documents themselves are associated to the system objects, or their data streams, by the system infrastructure through a system of identifiers and timestamps. Data streams are associated with their source through its system identifier and are time stamped as well. The identifiers along with the timestamps provide a simple mechanism for obtaining any system metadata required by any particular data or object reference, at any time.

In the MOOS design approach there are four general classes of objects: sensors, instruments, platforms and communication links. Each of these object classes may have specialized subclasses of objects with their own metadata requirements. As stated above there may be separate metadata requirements for the science uses of the data, for system configuration and management, and for operational use. All system objects are required to possess a unique system ID. The software system infrastructure design associates to every system object *metadata containers* to hold that object's metadata.

In this approach, a series of individual metadata object class specifications are defined, which, when combined, form the complete set of specifications of the metadata required for all types of system objects. Specifications are defined using XML schema. The specific metadata for a specific system object are then instances of schema, i.e.XML documents, which conform to the specification schema. To illustrate the complete prototype process we will review these steps for a relatively simple class of system object called a *sensor*. The first step is to design a schema that describes the metadata required for the class of sensor. In this case we will do this for sensors that might occur, for example, on a CTD. Using an XML design tool, such as XML Spy™, we could lay out the schema as shown in Fig. 2.

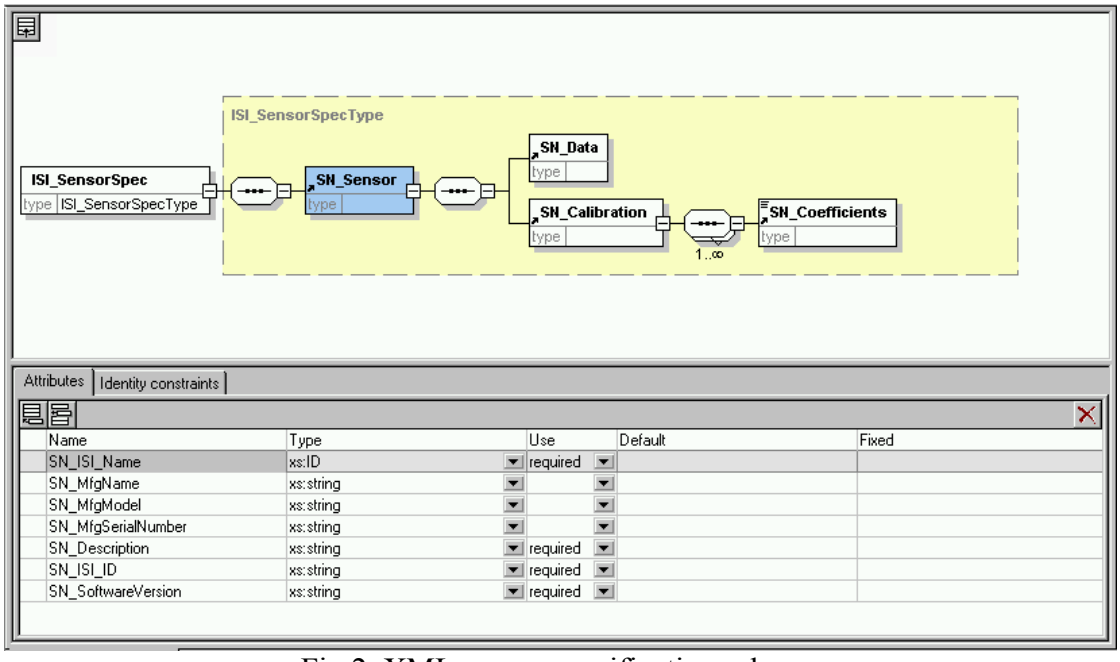

Fig 2. XML sensor specification schema

The root node *ISI_SensorSpec* is an ISI specification node. Its only attribute (not shown) is a locator field describing where it is filed or accessed over the web. This is followed by a *SN_Sensor* node, which has the following attributes as shown in Fig. 2.: *SN_ ISI_Name* , *SN_MfgName*, *SN_MfgModel*, *SN_MfgSerialNumber*, *SN_Description*, *SN_ISI_ID*, *ISI_SoftwareVersion*. This is followed by two nodes, *SN_Data*, and *SN_Calibration*. The calibration node is followed by a (possibly empty) list of *SN_Coefficients* nodes, consisting of name-value pairs. Each of the node elements has a list of attributes (shown only for *SN_Sensor* in Fig. 2).

Once the schema design is established, the next step is to produce an entry form application that can create and access XML documents that conform to the schema. In this prototype methodology this is done using the IBM VisualAge™ for JAVA IDE, and the EasyXML Bean Suite V4.0 available from the IBM Alpha Works developer site (IBM.ALPHA-WWW). This is primarily a matter of simply drawing a representation of the schema in tree form on a visual design surface, creating a user interface form using JAVA Swing or AWT bean elements for interacting with the XML beans representing the schema, and then 'wiring' the schema nodes and attributes to the interface elements. The visual composition process is shown in Fig. 3. The 'wires' describe how actions on the form bind to elements (nodes, or attributes) in the XML schema tree. The connections can be made two ways. If a XML document is retrieved, it is parsed by a SAX parser and the resulting events can cause information in the document to be used to fill in the form. Conversely events created by the user interacting with the form cause information in the form to be written to the nodes of the schema, and when saved, written out in syntactically correct XML

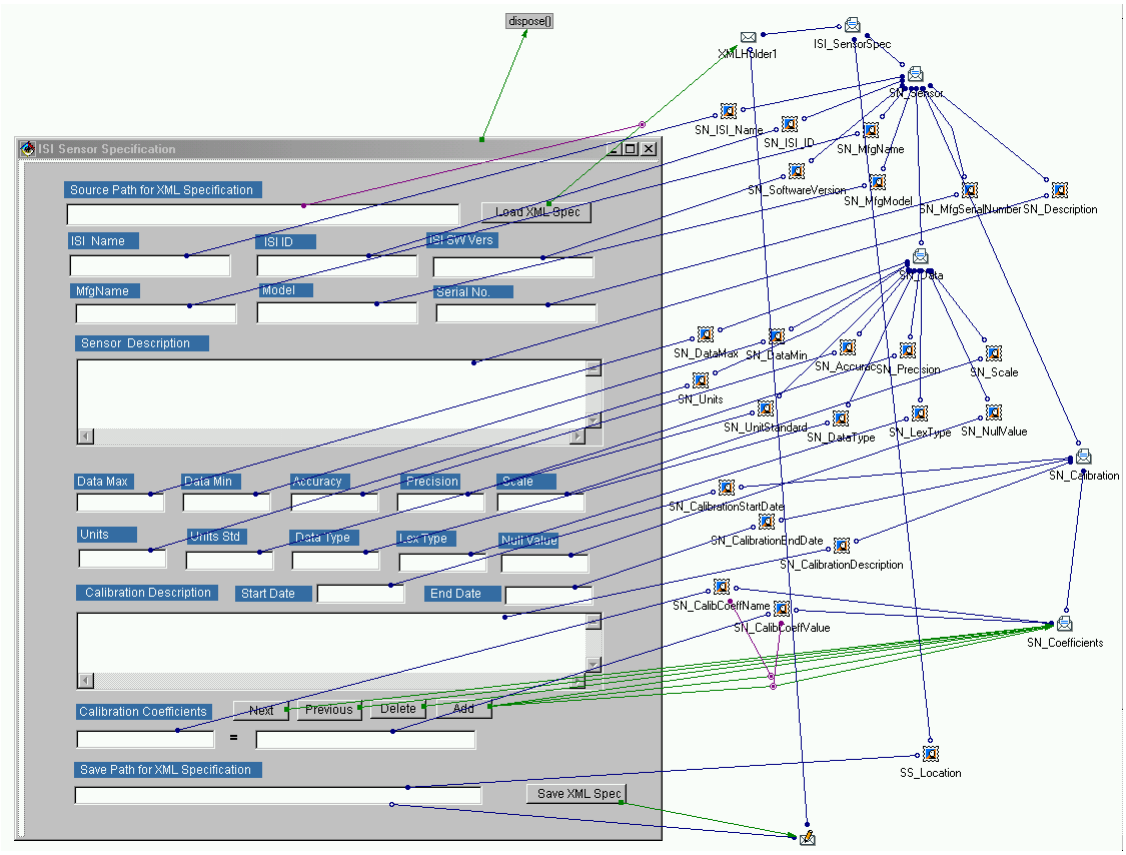

Fig 3 . User interface form design, and wiring diagram, for XML sensor schema

according to the schema. As indicated, the tree diagram to the right of the visual form corresponds exactly to the nodes and attributes of the nodes defined in the schema in Fig. 2.

Once the visual composition process is complete, it is used to automatically generate JAVA code that can be executed on its own, or in an applet within a web application (Fig. 4). Using the generated application, the user fills in the fields on the form shown by the application for sensor specifications, and when complete, the application will create an XML metadata document based on the XML schema (Fig. 5). As indicated, because of the two-way nature of the connections in the composition, the same application that creates an XML metadata document for a system object can also be used to read or revise one already written. Literally no JAVA code needed to be manually written to create this application. The entire process is visual.

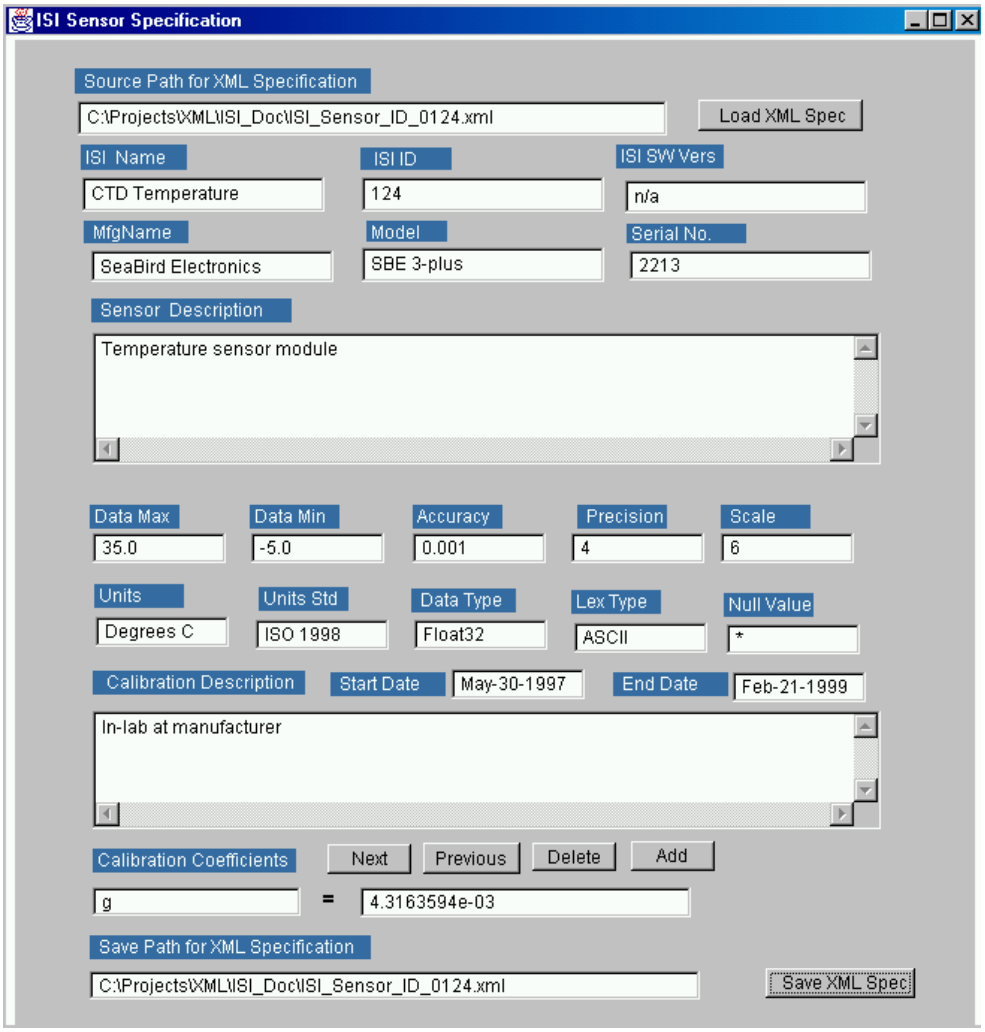

Fig 4. User interface form to create, view, and revise XML sensor metadata

When the application is executed, the form comes up with empty fields, unless default values are set in the form. The information that is required for each field is fairly self-explanatory to a user. Once the fields are filled out the user defines the locator reference in the 'Save path for XML Specification' edit field and then selects the 'Save XML Spec' button. The XML document is saved in correct XML (Fig. 5). The application automatically enters the same locator reference in the 'Source path for XML Specification'. If the user wishes to review or revise the saved file, it can

be loaded by selecting the 'Load XML Spec' button. In this way a collection of XML metadata documents can be created for all system elements. Although not shown here, a platform metadata schema and form has been prototyped that includes a list of the Sensor_ISI_ID, and file locators for the metadata documents for the sensors on that platform. In essentially what is an object-oriented methodology, the schema is a class definition, the XML metadata documents are instances of the class and the interface forms are create, view, or revise methods associated to the class. For this reason it is convenient to organize the information created and managed by these tools in an object-oriented directory structure. Each directory is associated to an object schema (a file of type 'xsd'), includes the creator/viewer/editor application corresponding to it, and all the XML metadata documents that conform to the schema and created by the application. This is just one simple way that a potentially large amount of metadata could be managed. The plan at MBARI is to create a catalog of metadata accessible within the database that manages the data, to provide access to the required metadata.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <ISI_SensorSpec SS_Location="C:\Projects\XML\ISI_Doc\ISI_Sensor_ID_0124.xml">
  - <SN_Sensor SN_Description="Temperature sensor module" SN_ISI_ID="124"
      SN_ISI_Name="CTD Temperature" SN_MfgModel="SBE 3-plus" SN_MfgName="SeaBird
      Electronics" SN_MfgSerialNumber="2213" SN_SoftwareVersion="n/a">
      <SN_Data SN_Accuracy="0.001" SN_DataMax="35.0" SN_DataMin="-5.0"
        SN_DataType="Float32" SN_LexType="ASCII" SN_NullValue="*" SN_Precision="4"
        SN_Scale="6" SN_UnitStandard="ISO 1998" SN_Units="Degrees C" />
    - <SN_Calibration SN_CalibrationDescription="In-lab at manufacturer"
        SN_CalibrationEndDate="Feb-21-1999" SN_CalibrationStartDate="May-30-1997">
        <SN_Coefficients SN_CalibCoeffName="g" SN_CalibCoeffValue="4.3163594e-03" />
        <SN_Coefficients SN_CalibCoeffName="h" SN_CalibCoeffValue="6.41530157e+04" />
        <SN_Coefficients SN_CalibCoeffName="i" SN_CalibCoeffValue="2.27235685e-05" />
        <SN_Coefficients SN_CalibCoeffName="j" SN_CalibCoeffValue="2.1715345e-06" />
      </SN_Calibration>
    </SN_Sensor>
  </ISI_SensorSpec>
```

Fig 5. XML sensor metadata document created by interface form

5.0 Conclusions

The main conclusion has already been widely recognized: XML technology is the appropriate choice for managing the metadata problem for ocean observing systems, as well as data exchange. We have also shown that it is relatively straightforward, using this technology, to design metadata specifications for classes of objects in an observing system and to design user friendly forms for creating, displaying, and revising metadata documents expressed in XML without requiring the user to read XML.

We have also illustrated a prototype process for developing metadata specifications, using XML schema, and show how users can create, access, view and revise metadata documents in XML.

We have also suggested that an object-oriented approach to metadata design, use and management, using object-oriented principles for defining a class object hierarchy, with associated methods and instances, may be an appropriate metaphor. In this metaphor, metadata specifications become object class definitions, and are expressed using XML schemas. The forms for creating, viewing, revising metadata document for each metadata specification class become methods associated to the class, and are attached to the class. Specific metadata documents that describe specific sensors, instruments, platforms, and communication links are simply *instances* of the metadata specifications for these classes of system objects.

Acknowledgements
We wish to acknowledge our appreciation to the David and Lucille Packard Foundation for their generous support of research and development at MBARI, of which this effort is a part.

References

Malone, T.C., and Cole, M.:2000, Towards a global scale coastal ocean observing system, *Oceanography*, 13, 7-11.

Nowlin, W.D. et al, 1996, An observing system for climate. *Bulletin of the American Meteorological Society*.,77,2243-2273.

O'Reilly, Tom, Duane Edgington, Daniel Davis, Richard Henthorn, Michael P. McCann, Timothy Meese, Wayne Radochonski, Michael Risi, Brent Roman, Rich Schramm, 2001: "Smart Network" Infrastructure for the MBARI Ocean Observing System, *Oceans IEEE 2001 Conference Proceedings*, Honolulu, Hawaii - Nov. 5-8.

Web Sites

GOOS-WWW: http://ioc.unesco.org/goos/ (current November, 2002)
IBM.ALPHA-WWW: http://www.alphaworks.ibm.com/ab.nsf/bean/easyXML/ (current November, 2002)
MEDI WWW: http://ioc.unesco.org/medi/ (current November, 2002)
MARINE XML-WWW: http://ioc.unesco.org/marinexml/ (current November, 2002)
MBARI.ORG-WWW: http://www.mbari.org (current November, 2002)
OCEAN.US-WWW: http://www.ocean.us.net/home.jsp (current November, 2002)
XML.COM-WWW: http://www.xml.com/ (current November, 2002).