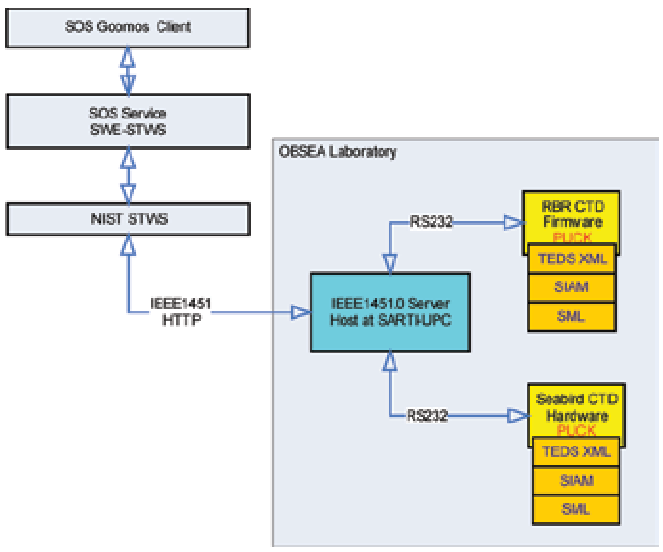

Fig.11 - Block Diagram of the Test Bench

Conclusions

PUCK Protocol can co-exist and it is compatible with other existing standards as IEEE1451 or SWE – SOS. The use of PUCK protocol with in an instrument facilitate the integration of the instrument within an observatory allowing storage of the description of the instrument metadata in different payloads types as IEEE1451 XML TEDS or SensorML. The engineering effort required integrating a PUCK enable instrument into and observatory is very small. Within a working day a computer science engineer is able to understand and communicate with a PUCK enable instrument, storing and configuring its payload. Approximately one week is enough time to define the payload and generate the code to be ready to integrate the instrument into the observatory. An automatic instrument recognition protocol has been proposed in order to enable the host to automatically configure a new instrument using PUCK Protocol and different Payload types.

Fig.13 - SOS Client from Compusult

References:

[1] Lee, K., "IEEE 1451: A Standard in Support of Smart Transducer Networking", Proceedings of the 17th IEEE Instrumentation and Measurement Technology Conference, Baltimore, MD, May 1-4, 2000, Vol. 2, p.525-528

[2]George Percivall, Carl Reed "OGC® Sensor Web Enablement Standards", Sensors & Transducers Journal, Vol.71, Issue 9, September 2006, pp.698-706, ISSN 1726-5479

[3] Kent L. HEADLEY, Dan DAVIS, Duane EDGINGTON, Lance McBRIDE , Thomas C. O'REILLY, Michael RISI "Managing Sensor Network Configuration and Metadata in Ocean Observatories Using Instrument Pucks" Third International Workshop on Scientific Use of Submarine Cables and Related Technologies, 25-27 June 2003

[4] Marc Nogueras, Carola Artero, Joquín del Rio, Antoni Mànuel, David Sarrià, "Control and acquisition system design for an Expandable Seafloor Observatory", IEEE OCEANS09, 11-14 May, Bremen, Germany

[5]M. Nogueras, J. Santamaria, A. Mànuel, "Construction of the OBSEA cabled Submarine Observatory" Instrumentation Viewpoint, Num.6, pp33-34, autumn 2007.

[6] T.O'Reilly, K.Headley et al, "MBARI technology for self-configuring interoperable ocean observatories", Proceedings of the Marine Technology Society / Institute of Electrical and Electronics Engineers Oceans Conference, Boston, Massachusetts

[7] Rueda, C., Bermudez, L., Fredericks, J. The MMI Ontology Registry and Repository: A Portal for Marine Metadata Interoperability. MTS/IEEE Oceans'09. Biloxi, Mississippi. October, 2009.

Excerpts from "Instrument Interface Standards for Interoperable Ocean Sensor Networks"

T.C. O'Reilly[1], K. Headley[1], D.R. Edgington[1], C. Rueda[1], K. Lee[2], E. Song[2], J. Zedlitz[3], J. del Rio[4], D. Toma[4], A. Manuel[4], E. Delory[5], C. Waldmann[6], S. Fairgrieve[7], L. Bermudez[8], E. Bridger[9], P. Bogden[9], A. Amirault[10]

[1]Monterey Bay Aquarium Research Institute (oreilly@mbari.org), [2]National Institute of Standards and Technology (kang.lee@nist.gov), [3]Christian Albrechts University at Kiel (jze@informatik.uni-kiel.de), [4]Polytechnic University of Catalunya (joaquin.del.rio@upc.edu), [5]dBscale (eric.delory@dbscale.com), [6]University of Bremen (waldmann@marum.de), [7]Northrop Grumman Corporation (Scott.Fairgrieve@ngc.com), [8]SURA (bermudez@sura.org), [9]GoMOOS (eric@gomoos.org), [10]Compusult Ltd (angela@compusult.net).

Transducer Electronic Data Sheet (TEDS)

The transducer electronic data sheet ("TEDS") is a key concept of IEEE 1451. A TEDS describes characteristics and

capabilities of components such as transducers, TIMs, and communications links in a standard way. Applications can retrieve the TEDS through the IEEE 1451 protocols to dynamically discover instruments, sensors, and actuators as well as other system characteristics.
IEEE 1451.0 defines the TEDS formats for the family of IEEE 1451 standards. The IEEE 1451.0 TEDS are classified into mandatory and optional TEDS. The mandatory TEDS include Meta TEDS, Transducer Channel TEDS, PHY TEDS, and User Transducer Name TEDS. The optional TEDS include Calibration TEDS, Frequency Response TEDS, Transfer Function TEDS, Manufacturer-defined TEDS, End User Application-specific TEDS, and Text-based TEDS, which include Meta ID TEDS, Transducer Channel ID TEDS, Calibration ID TEDS, Command TEDS, Location and Title TEDS, and Geo-location TEDS.

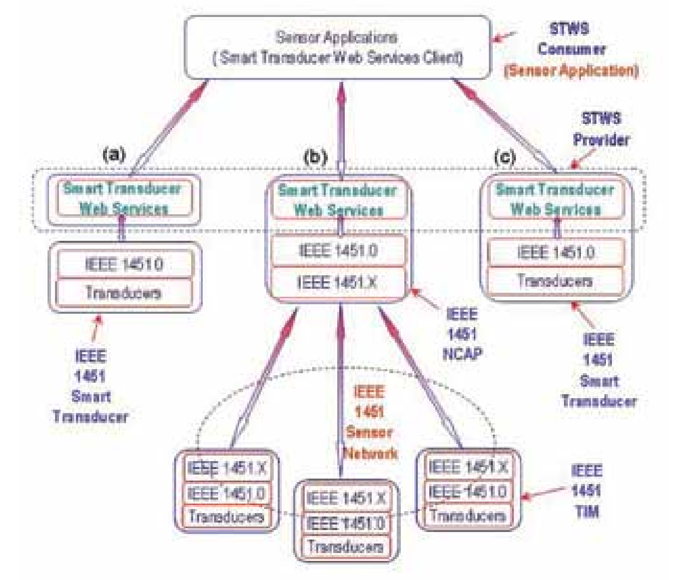

Fig.14 - STWS unified web service for IEEE 1451 smart transducers.

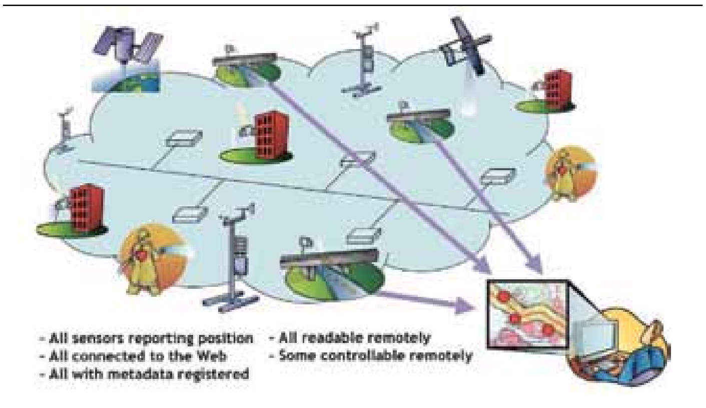

Fig.15 - Sensor web (courtesy of OGC).

Smart Tranducer Web Service (STWS)

The STWS consists of a set of web services for accessing IEEE 1451 smart transducers. The STWS described in Web Service Definition Language (WSDL) is based on service-oriented architecture (SOA) and the IEEE 1451.0 transducer services. The STWS WSDL specification is divided into six major elements: definitions, types, messages, portType, binding, and service. The STWS provides a unified Web service for IEEE 1451 smart transducers. The STWS component could reside in a separate computer to serve an IEEE 1451 smart transducer as shown in section (a) of Figure 14. It can reside in an NCAP to serve an IEEE 1451-based sensor network as shown in section (b) of Figure 14. The STWS component could also reside in an integrated IEEE 1451 smart transducer as shown in section (c) of Figure 14. The STWS provides a standard way to achieve interoperability of IEEE 1451 smart transducers with sensor applications.

OGC Sensor Web Enablement

A sensor web (Figure 15) refers to Web-accessible sensor networks and archived sensor data that can be discovered and accessed using standard protocols and Application Program Interfaces (APIs).

The Open Geospatial Consortium - Sensor Web Enablement (OGC-SWE) group is building a framework of open standards to exploit Web-connected sensors and sensor systems, such as flood gauges, air pollution monitors, stress gauges on bridges, satellite-borne earth imaging devices, oceanographic instruments, and other sensors and sensor systems. The OGC-SWE initiative focuses on developing a set of standards to enable the discovery, exchange, and processing of sensor observations and tasking of sensor systems. The OGC-SWE members have developed and tested the following specifications:
• Observations and Measurements (O&M) – Standard conceptual model and XML schema to encode observations and measurements. O&M defines an “observation” as an event whose result is an estimate of the value of some property of a feature of interest, obtained using a specified procedure. A sensor, channel, or systems of sensors could all be treated as a procedure. Data interoperability between instruments can be achieved with the O&M standard.
• Sensor Model Language (SensorML) – Standard conceptual model and XML schema to describe sensors, systems, and processes; provides information needed for discovery of sensors, location of sensor observations, configuration of sensor networks, processing of low-level sensor observations, and listing of ”task-able” processes
• Transducer Markup Language (TransducerML or TML) – Conceptual model and XML schema to describe transducers and real-time streaming of data to and from sensor systems.
• Sensor Observation Service (SOS) - Standard web service interface for requesting, registering, filtering, and retrieving observations and sensor system information.
SOS is the intermediary between a client and an observation repository or near real-time sensor channel.
• Sensor Planning Service (SPS) – Standard web service interface for requesting user-driven acquisitions and observations. SPS is the intermediary between a client and a sensor collection management environment.
• Sensor Alert Service (SAS) – Standard web service interface for publishing and subscribing to alerts from sensors.
• Web Notification Service (WNS) – Standard web service interface for asynchronous delivery of messages or alerts from SAS and SPS web services and other elements of service workflows.

SensorML

SensorML is a key component of SWE, providing standard sensor models and an XML encoding to describe any process associated with a sensor. All processes define their inputs, outputs, parameters, methods, and relevant metadata. SensorML models detectors and sensors as processes that convert real phenomena to data. It provides a functional model of a sensor system, rather than a detailed description of its hardware. It also treats sensor systems and the system's components (e.g., sensors, actuators, platforms, etc.) as processes. Thus, each component can be included as a part of one or more process chains that can either describe the lineage of the observations or provide a process for geo-locating and processing the observations to higher level information. In addition, SensorML provides additional metadata that are useful for enabling discovery, identifying system constraints, providing contacts and references, and describing "taskable" properties, interfaces, and physical properties.

Integration of IEEE 1451 and OGC-SWE

While the IEEE 1451 suite of standards deals with sensor metadata and sensor data from physical sensors to the network, OGC-SWE brings sensor information into Web applications. Applying both sets of standards will ultimately achieve the ease of use of sensors and ability to transfer sensor information from physical sensors to applications in a seamless manner using consensus-based standards. The question is how to apply or integrate IEEE 1451 and OGC-SWE to achieve instrument interoperability. The STWS is the proposed method to seamlessly integrate IEEE 1451 standards with the OGC-SWE standards and other sensor applications.
The OGC Web Services 5 interoperability exercise focused on integration of SWE interfaces and encodings into workflows to demonstrate the ability of SWE specifications to support operational needs. OWS-5's "Team-1451" implemented and demonstrated the integration of IEEE 1451-based smart sensors and the SWE Web Services through the STWS.

Interoperable Test-Bed

Description of Interoperable Test-Bed

We have developed an interoperable instrument test-bed in collaboration with OGC and some members of the Sensor Standards Harmonization Working Group (SSHWG) led by the National Institute of Standards and Technology (NIST). The goal of this effort is to demonstrate how IEEE 1451, OGC Sensor Web Enablement, and MBARI PUCK protocols can be integrated to rapidly acquire, fuse, and assess data from a diverse set of instruments and individual observatories.
The test-bed was originally demonstrated at the 2008 Ocean Innovations Interoperability Workshop and has since been refined and most recently demonstrated at the 2009 NSF Ocean Observing Initiative Instrumentation Workshop.

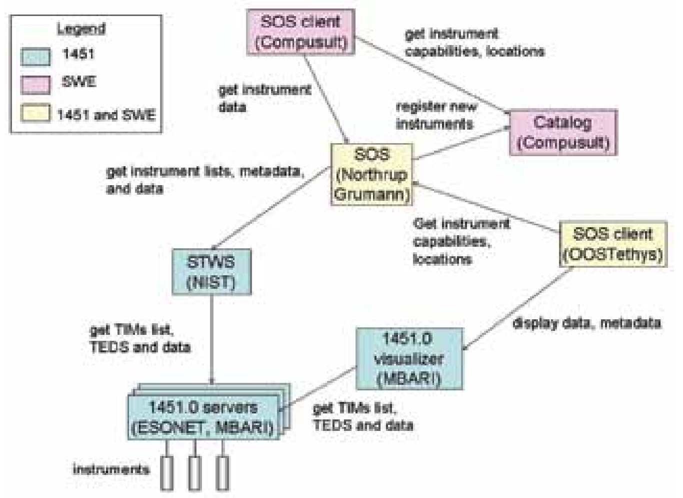

Fig.16 - Interoperable instrument test-bed architecture.

The test-bed currently integrates individual observatories at four different institutions in the USA and Europe into a single sensor network (Table II). Three of these observatories are associated with the European Seafloor Observatory Network (ESONET) and are located in Spain and Germany.
The fourth is located at the Monterey Bay Aquarium Research Institute (MBARI) in California USA.
Each individual observatory contains multiple instruments and independently-developed software components, some of which do not conform to recognized standards. However, team-members at each observatory have implemented IEEE 1451 "adapter" software that maps between IEEE 1451.0 protocol and their observatory protocols. Thus Internet applications that recognize IEEE 1451.0 can access the observatories' instruments through an IEEE 1451.0 server associated with each observatory (lower left corner of Figure 16).

Table II. Test-Bed Observatories and Instruments

Observatory	Instrument
UPC-SARTI (Vilanova, Spain)	SBE37-SM CTD
University of Bremen (Germany)	Sea and Sun CTD SBE37-SM CTD
Christian Albrechts University at Kiel (Germany)	Sea and Sun CTD IFM GeoMar meteorological instruments
MBARI (USA)	SBE37-SM (w/PUCK) RBR XR420 CTD (w/PUCK) WETLabs Triplet fluorometer (w/PUCK) ASIMET wind sensor

One such application is the STWS, which provides a bridge between OGC-SWE protocol and IEEE 1451.0.
Thus OGC-SWE components such as a Sensor Observation Service (SOS) can access the individual observatory instruments through the STWS.
The current test-bed utilizes just a subset of IEEE 1451.0, including methods to discover TIMs on each NCAP, retrieve various TEDS, get the geo-location of each TIM, and acquire data from each TIM channel.

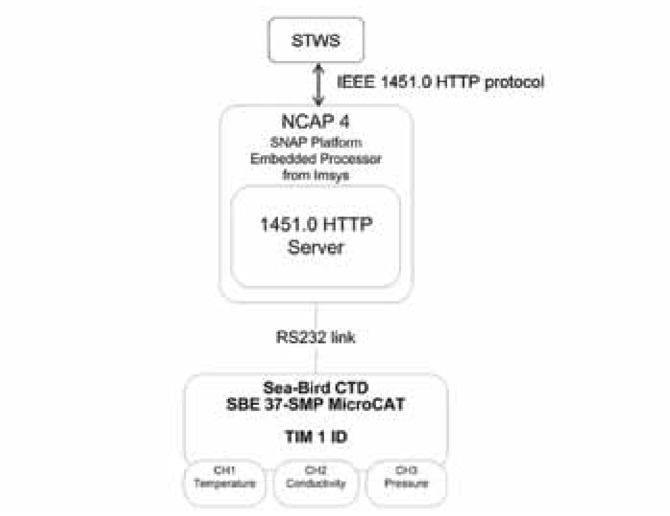

Fig.17 - UPC-SARTI observatory architecture.

Figure 17 schematically depicts the UPC-SARTI observatory which is a prototype for the OBSEA cable-to-shore observatory to be deployed in the western Mediterranean Sea in 2010. The UPC-SARTI NCAP executes the IEEE 1451.0 server and instrument drivers. One or more RS-232 instruments are plugged into the NCAP. The server and instrument drivers treat the attached instruments as IEEE 1451 TIMs, each TIM containing one or more transducer channels.
The NCAP is implemented by a very low-power yet capable Imsys Technologies SNAP module and the server and drivers are implemented in Java J2ME. The server can be reconfigured as instruments are installed or removed from the NCAP. For demonstration purposes, the observatory's Seabird CTD sensor can be installed in a hyperbaric chamber to simulate varying water depth. The basic UPC-SARTI design is heavily influenced by IEEE 1451.
In contrast the MBARI observatory provides an example of a "legacy" system. Over the past several years MBARI has developed and deployed observatory middleware called "SIAM" for use on its moored observatories (MOOS) as well as the MARS cable-to-shore observatory.
For each physical instrument in the observatory, SIAM provides an instrument driver that presents a generic "service" interface to clients on a TCP/IP network. Similar in design philosophy to IEEE 1451, the SIAM instrument service interface comprehensively defines how clients configure, control, and retrieve data from the associated instrument.
Clients request these operations through the generic interface, without having to know the native instrument serial protocol, as those details are "hidden" by the SIAM instrument drivers. SIAM also recognizes MBARI PUCK protocol and automatically retrieves driver code and instrument metadata from an instrument that implements PUCK, thus achieving "plug and work" behavior.
SIAM has proven very useful and extensible over the past several years. The interoperability test-bed thus presents an excellent opportunity to integrate a "legacy" system (SIAM) with standard interfaces (IEEE 1451, OGC-SWE).
In our test-bed the SIAM software executes on a MBARI Mooring Controller (MMC), which implements an IEEE 1451 NCAP (Figure 18). Instruments are plugged into serial ports on the MMC.
The MBARI IEEE 1451.0 server runs on a workstation host on the MBARI network and communicates with the SIAM instrument services via Java Remote Method Invocation (RMI).

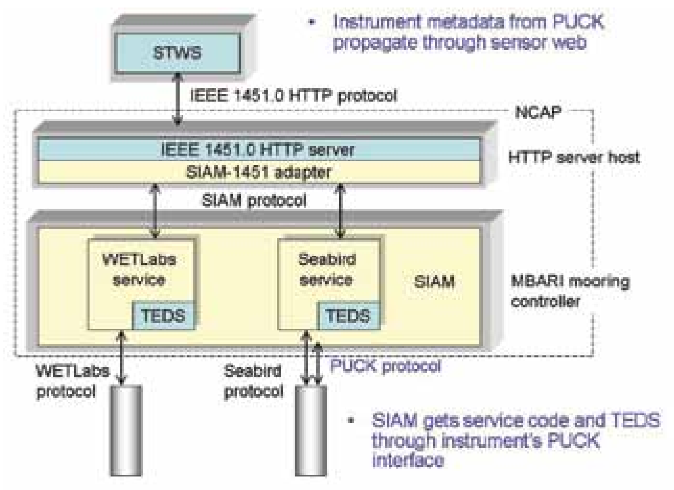

Fig.18 - MBARI SIAM observatory adapted to IEEE 1451.0.

To integrate SIAM with IEEE 1451.0, the MBARI developers implemented an "adapter" component that maps between IEEE 1451.0 requests coming through the server and methods in the SIAM service interface (Figure 18). Given the similar design philosophy between SIAM and IEEE 1451.0, the mapping between the two protocols is straightforward.
Thus when the IEEE 1451.0 server receives a client request, it issues appropriate method calls to the SIAM instrument service(s), transforms the values returned by the instrument service to IEEE 1451.0 format, and returns the values to the IEEE 1451.0 client.
The MBARI developers also slightly extended the SIAM instrument service classes to incorporate the IEEE 1451 "transducer" concept.
SIAM recognizes PUCK protocol and retrieves the payload of a PUCK-enabled instrument when the device is plugged in. For the test-bed, metadata needed by the various IEEE 1451 TEDS are stored in the instruments' PUCK payload, automatically retrieved by SIAM when the instrument is installed, and transferred to IEEE 1451.0 clients (including the STWS) on request. Thus metadata retrieved from the i strument itself through PUCK protocol is propagated across the sensor web.

Test-bed Performance

The current test-bed relies on polling between SWE and IEEE 1451 components to update states across the system. This is because the current SOS v1.0 specification does not support asynchronous operations; instead, the SOS relies on a typical web service HTTP POST request/response mechanism for retrieving sensor information and observations. This approach can be quite inefficient, as reflected in sometimes sluggish performance of SWE clients. Asynchronous event notification between IEEE 1451 and SWE would enable much more timely and efficient update of SWE clients when instruments are installed into the network, change their position, acquire a sample, or otherwise change in an asynchronous way.
Existing OGC-SWE specifications such as the Sensor Alert Service and Web Notification Service support asynchronous notifications, and the OGC is investigating ways of incorporating asynchronous operations in future versions of the SOS and other OGC specifications.
The current SOS implementation relies on many requests to the STWS in order to retrieve the latest information regarding available IEEE 1451 sensors.
Information from the STWS is used to populate the SOS Capabilities document, which tells SOS clients what sensors and observations are available, as well as SensorML documents describing the sensors themselves and O&M observations describing the data coming from those sensors.
The current implementation employs caching of many of the STWS responses in order to maximize efficiency, but more effective caching can be added to further improve efficiency. Caching can also be used on the SOS client to minimize the number of new requests that need to be made to the SOS in order to discover and describe sensors.
The current test-bed utilizes high-speed network links throughout. A more realistic design will incorporate low-bandwidth intermittent links to simulate satellite communications for moorings and perhaps acoustic links for underwater applications.

IEEE 1451 TEDS and SensorML

IEEE 1451 and OGC-SWE each provide a metadata framework to describe the characteristics of sensors. IEEE 1451 TEDS focus primarily on physical characteristics of sensors, instruments, and communication links which are closely associated with TIMs, whereas SensorML is applicable to high-level applications. SensorML provides a more comprehensive model that includes complex characteristics such as sensor data processing procedures and data acquisition schedules. Individual observatories in the current test-bed have no explicit notion of OGC-SWE and provide only TEDS to the IEEE 1451.0 layer. (The TEDS are subsequently mapped to basic SensorML elements by the Northrop Grumann SOS).
However, the additional sensor information provided by SensorML (but apparently not by TEDS) can be extremely valuable in a broader sensor web. Hu et al describe a TEDS-to-SensorML mapping scheme but also point out the complexities and limitations of their approach. An alternate approach could add a method to transfer "opaque" metadata through IEEE 1451.0. In our case, the observatory could transfer a SensorML document by this method.
In any case, the TEDS-to-SensorML integration problem requires more research.

Additional Functionality

The test-bed currently emphasizes data interoperability. As a next step we plan to demonstrate the capability to configure and operate instruments through a standard Internet interface.
This step will require integration and perhaps modification of the Sensor Planning Service and IEEE 1451 standards.
Thus far, the test-bed implements only a few methods in the IEEE 1451.0 standard; we plan to add more functionality in the future. Most of the test-bed instruments return raw data with a fixed and simple format that is easily mapped to the IEEE 1451.0 standard data format. We also plan to integrate instruments, such as acoustic doppler current profilers (ADCP) that generate more complex data structures.

CAN standards

Controller Area Network (CAN) was originally developed as a bus architecture for automobiles, but today is used in a wide variety of applications. The CAN-bus network provides a very efficient and robust platform for deterministic real-time applications of distributed sensors and actuators. Key advantages provided by CAN-bus include robust and efficient error detection and message transmission protocols. CAN-bus is based on OSI Reference Model layers 1 and 2 (physical and data link layers) and is standardized in ISO 11898.
Several application-level standards have been developed to run on CAN-bus, notably the CANopen communication protocol and device profile specification.
Several oceanographic applications that use CAN-bus and CANopen for onboard communications have been implemented, including autonomous underwater vehicles and buoys, and at least one manufacturer supplies oceanographic instruments for CAN-bus. We would like to investigate the use of CAN standards for future systems as well.
CANopen "device profiles" have been specified for several kinds of devices, including sensors and actuators. Every CANopen device profile specifies an "object dictionary" that describes all parameters and variables of that device. Objects can be simple data-types such as bytes, integers, floating point values, and strings, but also more complex data types like arrays. Some dictionary objects are mandatory, others are optional. The object dictionary is stored in a TEDS-like electronic data sheet.
CANopen bears conceptual similarities to IEEE 1451. For example, in addition to the TEDS-like device profiles, CANopen's "CAN-master" component is responsible for managing network communications between devices and the network, similar to the IEEE 1451 NCAP.
Unfortunately the IEEE 1451.6 CAN-bus working group is no longer active.
Nevertheless we could explore integration of CAN with OGCSWE standards, e.g., by "mapping" CANopen electronic data sheets to SensorML.