PUCK PROTOCOL USE CASE AT OBSEA. PUCK IS THE NEWEST OGC STANDARD FOR INSTRUMENT IDENTIFICATION AND COMMUNICATION

Joaquin del Rio, Daniel Mihai Toma, Thomas C. O'Reilly, Arne Bröring, David R. Dana, Felix Bache, Kent L. Headley, Antoni Manuel, Duane R. Edgington
SARTI Research Group. Electronics Dept. Universitat Politècnica de Catalunya (UPC).
Rambla Exposició 24, 08800, Vilanova i la Geltrú. Barcelona. Spain.+(34) 938 967 200
www.cdsarti.org

Abstract
Ocean observing systems may include a wide variety of sensor and instrument types, each with its own capabilities, communication protocols and data formats. Connecting disparate devices into a network typically requires specialized software drivers that translate command and data between the protocols of the individual instruments, and that of the platform on which they are installed. In addition, such platforms typically require extensive manual configuration to match the driver software and other operational details of each network port to a specific connected instrument. "plug-and-work" interoperability, using standardized protocols reduce the amount of instrument-specific software and manual configuration required for connecting instruments to an observatory system. First, the Sensor Interface Descriptor (SID) model, based on the Open Geospatial Consortium's (OGC) SensorML standard, to describe each instrument's protocol and data format, and to provide a generic driver/parser. Second, a new OGC standard known as PUCK protocol enables storage and retrieval of the SID file from the instrument itself. We demonstrate and evaluate this approach by applying it to three commonly used marine instruments in the OBSEA observatory test bed.

I. IMPLEMENTATION

We implemented a prototype plug-and-work observatory based on the recent new OGC standard PUCK Protocol and SID. The prototype was demonstrated at the general assembly meeting of the ESONET EU funded project in December 2010 in Marseille (France) [3]. This text is an excerpt of a most detailed publication available at [5].

A. Prototype Architecture

Figure 1 shows the main components of the prototype architecture. The objective of the prototype is to access data generated by instruments (1,2,3), connected to a host controller (5) through standard Internet clients (6,7) without any manual configuration steps at installation or run time. Instruments 2 and 3 are PUCK-enabled serial devices; the SID describing the instrument protocol and data formats is stored within the PUCK payload (8) of its corresponding instrument. The host controller (5) is implemented by an Internet-connected Windows laptop computer, and hosts several software components:
9a) A PUCK detector is assigned to each serial port, issuing a PUCK "soft break" at different baud rates until it receives a PUCK response from an attached instrument. If the detector determines that the instrument is newly installed based on its UUID, the SensorML and SID are either retrieved from the instrument's PUCK payload or from the SID repository, and a new SID Interpreter is created (9b). The SID Interpreter executes the initialization and sampling protocol declared in the SID. Data coming from the instrument will be parsed with the data format declared by the SID and processed (if necessary). These data can be sent to higher-level data management components on the network.

Our prototype uses the 52°North SOS (Sensor Observation Service) (11) and SOS client (6) to distribute the data. SOS server and client components are available as open source software from http://52north.org/sos. The data are also sent to a Data Turbine Ring Buffer [1 , 2] (12), for real-time data access through the Internet using the RDV Client (7).
The instrument SID files are generated beforehand with the SID

Instrument	Link Layer	PUCK Enabled	Provides PUCK payload	Data format used	Link to SID
Seabird SBE-16+ CTD on Smart Sensor Board	Ethernet	Via Smart sensor board	Via Smart sensor board	comma-delimited ASCII	http://52north.org/communities/sensorweb/examples/2011-07-28-Seabird-SBE-16P-3out.xml
Seabird SBE-37SM	RS232	Yes	Yes	comma-delimited ASCII	http://52north.org/communities/sensorweb/examples/2011-03-18-Seabird-SBE-37.xml
HOBI Labs HydroScat-2	RS232	No	No	comma-delimited and fixed-width ASCII	http://52north.org/communities/sensorweb/examples/2010-11-18-Hobilabs-HydroScat.xml
WET Labs ECO Triplet	RS232	Yes	No	space-delimited ASCII	http://52north.org/communities/sensorweb/examples/2010-11-10-WETlabs-Triplet.xml
RBR XR-420 CTD	RS232	Yes	Yes	space-delimited ASCII	http://52north.org/communities/sensorweb/examples/2011-03-18-RBR-xr420.xml

Table 1: Instruments used with interoperability prototype and SID URLs