OGC SWE-based Data Acquisition System Development for EGIM on EMSODEV EU Project

Daniel M. Toma, Joaquin del Rio, Javier Cadena, Ikram Bghiel, Enoc Martínez, Marc Nogueras
Universitat Politecnica de Barcelona
UPC- SARTI
Barcelona, Spain
Rambla Exposicion 24, 08800

Óscar Garcia, Juanjo Dañobeitia, Jordi Sorribas, Raquel Casas
Unidad de Tecnología Marina -CSIC
Barcelona, Spain
P. Marítimo de la Barceloneta 37-49, 08003

Jaume Piera, Rafael Bartolome, Raúl Bardaji
Instituto de Ciencias del Mar-CSIC
Barcelona, Spain
P. Marítimo de la Barceloneta 37-49, 08003

***Abstract*—The EMSODEV[1] (European Multidisciplinary Seafloor and water column Observatory DEVelopment) is an EU project whose general objective is to set up the full implementation and operation of the EMSO distributed Research Infrastructure (RI), through the development, testing and deployment of an EMSO Generic Instrument Module (EGIM). This research infrastructure will provide accurate records on marine environmental changes from distributed local nodes around Europe. These observations are critical to respond accurately to the social and scientific challenges such as climate change, changes in marine ecosystems, and marine hazards. In this paper we present the design and development of the EGIM data acquisition system. EGIM is able to operate on any EMSO node, mooring line, sea bed station, cabled or non-cabled and surface buoy. In fact a central function of EGIM within the EMSO infrastructure is to have a number of ocean locations where the same set of core variables are measured homogeneously: using the same hardware, same sensor references, same qualification methods, same calibration methods, same data format and access, and same maintenance procedures.**

Keywords— EMSO; data acquisition; EMSODE; EGIM; OGC; SOS; SE; SWE; Sensor; Zabbix.

I. INTRODUCTION

The general objective of the EMSODEV project is to implement a Generic Sensor Module (EGIM) within the EMSO (European Multidisciplinary Seafloor and water column Observatory). EMSO is a distributed infrastructure of strategically placed, deep sea and water column observatory nodes with the essential scientific objective of real-time, long-term monitoring of environmental processes related to the interaction between the geosphere, biosphere, and hydrosphere. The scientific drivers for developing and deploying the EGIM across a set of observatories in European Seas are manifold, spanning requirements to collect observations for understanding climate change, marine ecosystems, and geo-hazard early warning research. As illustrated in figure 1, the EGIM will utilize a comprehensive set of sensors and devices that meet particular technology readiness thresholds to collect observations including temperature, pressure, salinity, dissolved oxygen, turbidity, chlorophyll fluorescence, currents, and passive acoustics.

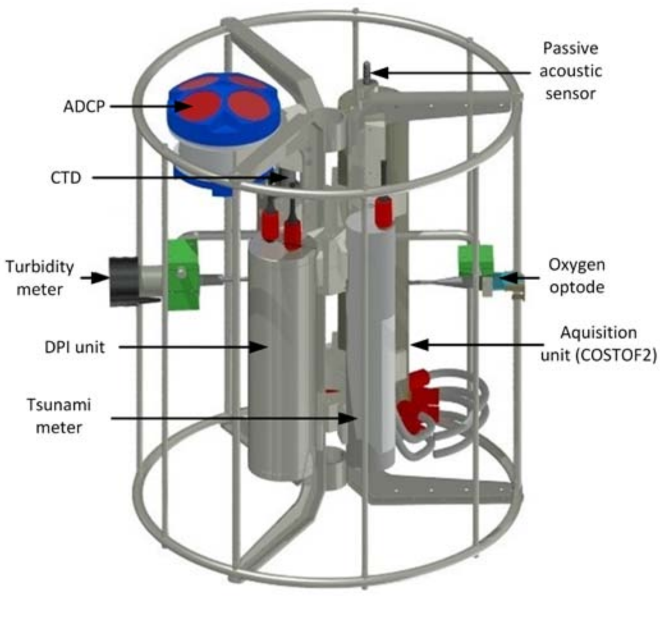

Fig. 1. EGIM prototype components

Relatively novel sensors will also be considered including those for pH, pCO2, and nutrients. Overall, this system will address the fullest possible set of Essential Climate Variables (e.g. from the WMO's GCOS-Global Climate Observing System program; www.wmo.int) at EMSO nodes.

Table 1. Core variables captured by the EGIM - EMSO Generic Instrument Module and their cross-disciplinary application

Variable	Geosciences	Physical Oceano-graphy	Biogeo-chemistry	Marine Ecology
Temperature	X	X	X	X
Conductivity	X	X	X	X
Pressure	X	X	X	X
Dissolved O_2	X	X	X	X
Turbidity	X	X	X	X
Ocean currents	X	X	X	X
Passive acoustics	X			X

EMSODEV, by means of EGIM, will provide unprecedented support for full standardization across EMSO. This is key to understanding regional scale phenomena. Data will be made coherent and attractive for the modeling community and for other potential stakeholders as shown in table 1. An open data policy has already been adopted in compliance with the recommendations being developed within the GEOSS initiative (The Global Earth Observation System of Systems). This allows the shared use of the data infrastructure and the free exchange of scientific information and knowledge. Our contribution to the implementation of the EGIM data acquisition system module (WP4 of the EMSODEV project) focuses on the development of a generic software for sensor web enablement. Through this generic software, the EGIM status data is directly inserted into a centralized SOS (Sensor Observation Service) server [2] and into a laboratory monitor system (LabMonitor) for recording events and alarms. Moreover, the software will be able to detect, register and start the data acquisition from any new sensors connected to EGIM. Based on this development, the project will set up a data management system enabling sensor management and data analysis compliant with the requirements of EU and international initiatives (e.g. EMODNET, GEOSS), and a state-of-art ICT (Information and Communications Technology) user environment.

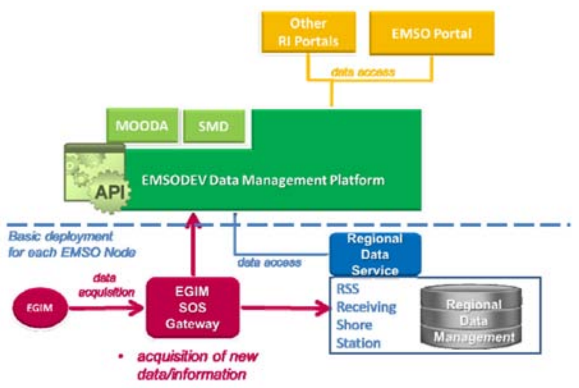

Fig. 2. EGIM diagram showing SOS Gateway relations

As shown in figure 2, the generic software for Sensor Web Enablement with the SOS server is located between the data source (EGIM) and the data management system in the EMSO Cyberinfrastructure (CI). The generic software for Sensor Web Enablement has two main functionalities. The first is to guarantee that the data is recorded properly from the EGIM hardware. The second is to register and insert the recorded data into a standardized Open Geospatial Consortium (OGC) SWE SOS [2] that works as a gateway for the EMSO data management system.

II. System Overview

The hardware required by the components of the EGIM acquisition system - the SOS server and the laboratory monitor system - has been implemented by virtualizing the hardware of the whole system, e.g. generating three virtual machines (‘Mussel’,‘SeaShell’ and ‘Donax’) for each separate roll. Each virtual machine has been configured with the necessary resources (database container, web server, VPN client ...) providing the necessary interfaces to communicate with the other hosts as shown in figure 3.

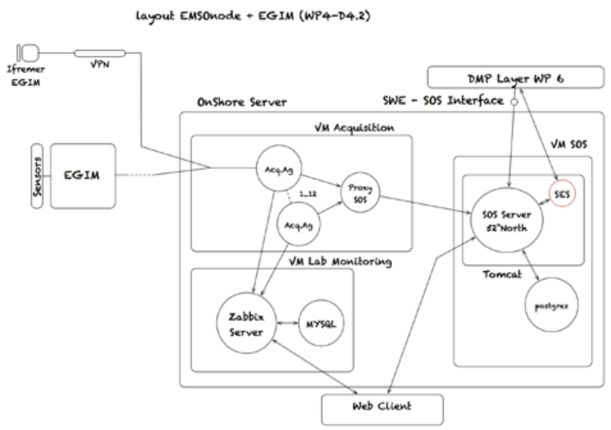

Fig. 3. General Server and Services Layout

III. SOS Gateway Interface

We analyzed the actual state-of-the-art SOS implementations, decided to use the 52°North SOS2.0 implementation to accomplish the SOS Gateway requirement. The 52°North SOS2.0 has capabilities to aggregate readings from live, in-situ and remote sensors. The service provides an interface to make sensors and sensor data archives both accessible through an interoperable web-based interface, using SensorML and Observation and Measurements (O&M).The main SOS 2.0 operations offered with this implementation are:

A. Core Extension

- *GetCapabilities*, for requesting a self-description of the service
- *GetObservatio*n, for requesting the pure sensor data encoded in Observations & Measurements 2.0 (O&M)
- *DescribeSensor* for requesting information about a certain sensor, encoded in a Sensor Model Language 1.0.1 (SensorML) instance document.

B. Enhanced Extension

- *GetFeatureOfInterest*, for requesting the GML 3.2.1 encoded representation of the feature that is the target of the observation.
- *GetObservaitonById*, for requesting the pure sensor data for a specific observation identifier

C. Transactional Extension

- *InsertSensor*, for publishing new sensors
- *UpdateSensorDescription*, for updating the description of a sensor
- *DeleteSensor*, for deleting a sensor
- *InsertObservation*, for publishing observations for registered sensors

D. Result Handling Extension

- *InsertResultTemplate*, for inserting a result template into a SOS server that describes the structure of the values of an *InsertResult* of *GetResult* request.
- *InsertResult*, for uploading raw values accordingly to the structure and encoding defined in the *InsertResultTemplate* request
- *GetResultTemplate*, for getting the result structure and encoding for specific parameter constellations
- *GetResult*, for getting the raw data for specific parameter constellations.

For this deployment we use the 'Mussel' virtual machine. We deployed the 52°North Web applications over a Tomcat Web server container version 7, configured with a Postgres database container. Some other small configurations for conditioning the application in our domain have been implemented.

In order to attend to client requests, we installed 52°North’s Helgoland Web client application[3] [4] to visualize the real time and historical data using the SOS Gateway as illustrated in figure 4. This web application has been opened for access from outside the local network

Fig. 4. Visualization of EGIM status data on 52°North Helgoland SOS Client

IV. Acquisition Services

The acquisition environment has been deployed on the 'Seashell' virtual machine. On this server all the elements are configured to accomplish the acquisition requirements. These requirements include the processes to acquire the data from the sensors that are connected to EGIM - the so-called acquisition agent - and to send the observations to the SOS Gateway and the Lab Monitor systems.

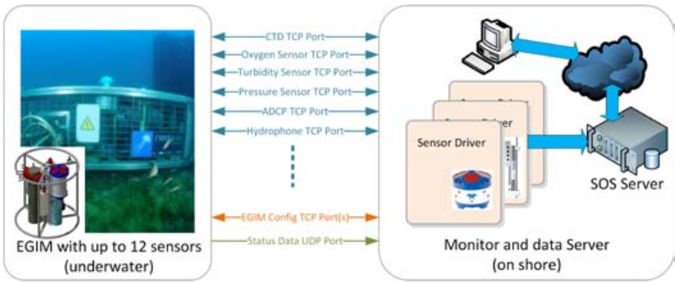

Fig. 5. EGIM functional description

To acquire data from the EGIM system, we need to distinguish between two kinds of reading procedures. First, there is a reading procedure for the external sensors connected to the EGIM system. The EGIM has the capability to host up to 12 sensors. Seven of these sensors are generic, as shown in figure 1. The five additional sensors provide additional ‘essential ocean variables‘, including chl-a, pCO2, pH, and photographic/video images. As illustrated in figure 5, these reading procedures are done based on TCP Socket connections. Second, there is a reading procedure of the EGIM internal sensors, which is done based on readings of UDP packets to a specific port. Once the agent reads these two kinds of data, we use a 'proxy SOS' tool to automatically executes all the data insert operations between the acquisition agent and the SOS server. Hence, this tool registers any new sensors connected to EGIM and sends the *InsertResult* queries for each new data acquired from EGIM. Moreover, the acquisition agent generates JSON requests to the Zabbix server [5], in order to add these values to the Lab Monitor's database.

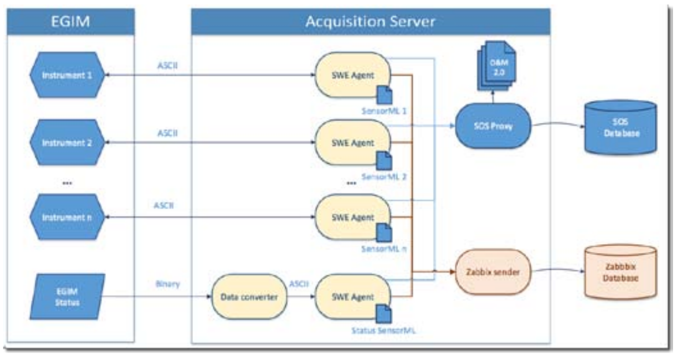

Fig. 6. Acquisition Components Overview

We have identified several categories of data shared between EGIM and CI. The following defines each one:

- Component descriptive data – Description of the platform/instrument configuration including instrument types, serial numbers, position of the deployment, calibration parameters.
- Command data – Commands and associated attributes such as when a command is scheduled to be executed.
- Instrument data – Data produced by the platform instruments, associated time tags, and attributes identifying the specific source instrument.
- Engineering data – Data describing the operational status of the system components.
- Metadata – Data describing the data. Metadata are data describing a resource, like an instrument or an information resource.

In order to provide the description of all these categories of data, we use the SensorML 2.0 standard. SensorML supports the ability to describe the components and encoding of real-time data streams, and to provide a link to the data stream itself [6]. This allows one to connect to a real-time data stream directly from a SensorML description and to use a generic data reader to parse the data stream. The act of describing a data stream into or out of a process (or sensor/actuator) is accomplished by having the input or output be of type *DataInterface*. The *DataInterface* element allows one to describe the *DataStream*, as well as provides an optional interface description.

The acquisition agent (the SWE agent) reads and decodes this file, which is encoded in an EXI format. It uses the decoded information to autoconfigure itself, which opens a communication port via an Ethernet connection with the instrument deployed by EGIM. This communication port has the capability to use both TCP and UDP protocols. The SWE agent starts getting information from the instrument in a push or pull mode. The data retrieved from the instrument is stored in XML files, following the *insertResult* format. This format is compliant with the Observation & Measurement Standard 2.0 and can be directly injected in the SOS database.

V. Monitor Lab

The benchmark test as well as the production processes require the visualization of some real data and historical trend data from sensors, with the objective to control some critical information by arranging triggers. In the same manner, we need to monitor the correct behavior of the whole system (EGIM Status data).

To accomplish this requirement, we built a LabMonitor system, using the Zabbix4 open source application. Zabbix is designed for monitoring availability and performance of IT infrastructure components. It works like a centralized monitoring system using active or passive agents for requesting or receiving from hosts. The system can use many protocols. In our scenario we use Zabbix agents to retrieve information about each virtual machine and observations from the EGIM system retrieved by the acquisition agents.

For this purpose, we have created the 'Donax' virtual machine. It uses a MySQL database as a data container and an Apache Web container for attending to Web client requests. In each server, a binary Zabbix agent that reports all information about the host to the Zabbix server has been installed. This functionality has also been added inside the acquisition software to send all the data acquired from the EGIM system. The acquisition agent gets the data received from EGIM sensors and sends it to the Zabbix server using a formatted JSON request. Then, the server informs the client if the data has been created successfully or it sends a report if any problem arises. Once the data has been received on LabMonitor, the data is written to the database, and can be visualized on the Web application. If a trigger has been configured for this data, the system will check the rule configuration and inform of any status change.

We can also check the state of the EGIM equipment by monitoring all the status values coming from EGIM status pack. We have configured alarms for critical data as the internal temperature, internal humidity, power consumption and water leak. In case any reporting alarm occurs, we set up an email account for receiving a message every time the trigger in the system switches on or off, informing about the sensor data involved and some more detailed information.

Finally, we set up a public access for remote monitoring purposes, which only require a web client for real time system data access. Moreover, it is also feasible to request historical data, which could be really useful for analyzing the events processes and crossing data.

VI. Conclusions

At the time of writing this document, we are in a development phase and compiling all the necessary components for the final production environment (estimated for October, 2016). As a result, the data may not have much meaning or may eventually contain some gaps on historical trends. New development in the following months may introduce changes, such as IP's, ports, etc..

We are trying to improve the system to complement the way that Data Management Platform (DMP) should receive the data. Initially, this configuration requires a connection polling to request data from the DMP to the SOS Server. This implies two operations for each request data. We have installed the Sensor Event Service in the SOS server. The objective is that users with a publish/subscribe-based interface could access to sensor data and measurements located at the SOS server. The SES basically produces notifications and provides methods to subscribe for notifications and retrieve the latest notification. Meanwhile, users can also register new sensors dynamically and send notifications to the service.

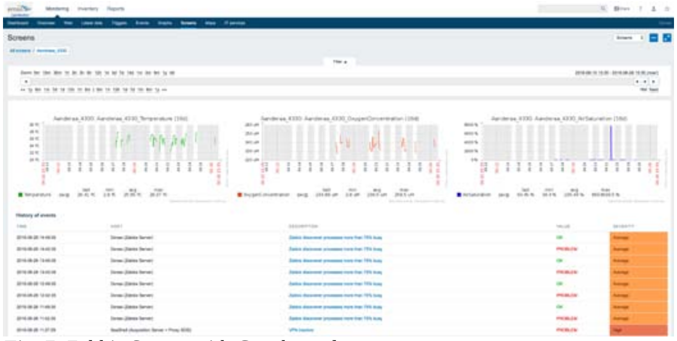

Fig. 7. Zabbix Screen with Graphs and events

ACKNOWLEDGMENT

This study benefited from the H2020 INFRADEV-3-2015 EMSODEV Project n°676555.

REFERENCES

[1] EMSO project site http://www.emso-eu.org/site/projects.html

[2] Arne Bröring, Christoph Stasch, Johannes Echterhoff “OGC® Sensor Observation Service Interface Standard”, http://www.opengis.net/doc/IS/SOS/2.0 , 2012-04-20

[3] 52 North SOS 2.0 implementation, http://52north.org/communities/sensorweb/sos/

[4] https://github.com/52North/helgoland

[5] Zabbix Monitoring System. http://www.zabbix.com/product.php

[6] del Río, J.; Mihai Toma, D.; O'Reilly, T.C.; Bröring, A; Dana, D.R.; Bache, F.; Headley, K.L.; Manuel-Lazaro, A; Edgington, D.R., "Standards-Based Plug & Work for Instruments in Ocean Observing Systems," Oceanic Engineering, IEEE Journal of , vol.39, no.3, pp.430,443, July 2014 doi: 10.1109/JOE.2013.2273277