ISTSOS: INVESTIGATION OF THE SENSOR OBSERVATION SERVICE

M. Cannata [a, *], M. Antonovic [a]

[a] Institute of Earth sciences, Dept. of Environment, Constructions and Design, University of Applied Sciences of Southern Switzerland, 6952 Canobbio, Switzerland - (massimiliano.cannata,milan.antonovic)@itu.edu.tr

Commission IV, WG IV/5

KEY WORDS: Sensor, Observations , Services, Internet,Web, Interoperability, standard, OGC

ABSTRACT:

Approving the new federal low about geo-information (RS 510.62), the Switzerland gave a great impulse to the establishment of a national spatial data infrastructure, which in agreement with the EU directive INSPIRE (2007/2/CE), shall be based on geo-services. This approach gives to data maintainers the freedom to choose the desired formats, software and data model but impose the provision of the required information throughout a well defined service. While some of the Geographical Web Services, like WMS and WFS, are today widely diffused and applied, others like SOS (Sensor Observation Service) are still a work in progress and need some revisions and verifications for their correct application.
The SOS primary aim is to provide a standardized service for accessing observations in a standard format. The secondary aim is to provide to intelligent sensor network an interface to interact with for automatic sensor registration and observation storage.
In order to investigate the maturity level of the version 1.0 of the SOS OGC standard, the IST (Institute of Earth Sciences) has developed a new software implementing this standard and has applied it for the management of the hydro-meteorological network of the Canton Ticino. The development of the new software, which is named istSOS, together with its application in a real case has provided the opportunity to evaluate some open issues, weakness and lacks that are discussed in this paper.
As a result the authors can overall conclude that the SOS v.1.0 is currently incomplete and still open to ambiguities but with some corrections it can easily be an invaluable resource for a great number of disciplines and actions, including climate changes, risk reduction, security improvement.

1. INTRODUCTION

1.1 General context of the research

In the last decades people's everyday life has been influenced by the fast development and diffusion of internet and its related technologies. Today we are fully into a new stage of the information technology: the "ubiquitous Web" era. While internet connection became faster and wider available, sensors and devices became smaller and cheaper: this trend is leading to the development and diffusion of Web Services (Cerami and St.Laurent, 2002) providing "on demand" information.
Taking into account this technological evolution, the Europe Union in 2007 emanates the INSPIRE directive (2007/2/CE) aiming to provide technical rules for the implementation of a European spatial data infrastructure able to support the definition of polices and the execution of activities impacting the environment. The Swiss Federal Government, in agreement with INSPIRE, emanates a new federal low about geo-information (RS 510.62) for the establishment of a national spatial infrastructure based on geo-services.
With this general view, also at the Cantonal level, local administrations are working in setting up geo-services for the provision of spatial information, including administrative and environmental data, to support decision making and execution of daily tasks.
In this framework, the Institute of Earth sciences (IST) has a Cantonal mandate for the management of the alarm system for the flooding from the Lake Maggiore (Cannata et al., 2005); it includes the management of the Canton Ticino hydro-meteorological monitoring network, the forecasting of lake levels, and the dissemination of alerts.
Taking into account: (i) the current regulations and their final aims together with (ii) the need of upgrading the entire system with new technologies (e.g.: GPRS communication modules for online sensors) and (iii) the fact that the flood forecasting requires the fusion of observations from other institutions, the IST has decided to investigate the promising Sensor Observation Service standard for the management and provision of sensor observed data.

1.2 Sensor Observation Service

In the geospatial context one of the most active body in the definition of Open Standards is the Open Geospatial Consortium, Inc.® (OGC, http://www.opengeospatial.org): an organization composed by more than 370 commercial, governmental, non-profit and research organizations worldwide working in close relationship with ISO/TC 211 (Geographic Information/Geomatics, http://www.isotc211.org). Some of the most known OGC standards, like Web Map Service, GML and Simple Features Access, have already become ISO standards while others more recent are still in an evolution phase (including verification, development of application, refinements, etc.). Within this framework, the Sensor Observation Service (SOS), available in its version 1.0 (Na and Priest, 2007) is a young standard with few real case applications and therefore that still requires verification.
The SOS has been created to enable the access to updated observation in a standardized way, taking full advantage of the

* Corresponding author. This is useful to know for communication with the appropriate person in cases with more than one author.

latest technology. This standard is part of the broader OGC Sensor Web Enablement initiative focusing on the real time integration of heterogeneous sensor with the scope to create a Web-connected network of sensors (Botts et al., 2007).
Like other OGC Web services, SOS is based on the exchange of requests and responses between the user (either data consumer of producer) and the service in a well known format as defined in the standard specifications. Requests are always identified by: *service*, *version* and *request* parameters (that in the case of SOS will have respectively the values "SOS", "1.0.0", and the desired request name) plus a series of other parameters that are specific for the request type.
While either XML (Extensible Markup Language) and KVP (Key-Value pairs) request are generally supported, responses are always provided as XML.
SOS specification envisages mandatory and optionally supported requests that are grouped into three profiles:

1. Core, mandatory requests enabling the exploration of the service, the description of sensors and the data gathering.
2. *Transactional*, optional requests supporting the acquisition of new data (new sensors and new observations).
3. *Enhanced*, optional miscellaneous requests for retrieving additional information or facilitate data access.

Within the *core* profile the *GetCapabilities* request allows to explore the service by retrieving information on: service provider, supported requests and respective parameters, offered procedures, observed properties and features of interest. The *DescribeSensor* request provides information on the sensor specified in the supported mandatory parameter *procedure*; its response include instruments capabilities, characteristics and other supporting data in the format of Sensor Model Language (SensorML,) (Botts and Robin, 2007) or Transducer Markup Language (TML) (Havens, 2007). The *GetObservation* request allows to obtain the desired observations, according a set of supported filtering and setting parameters, in the format of Observations and Measurements (O&M) (Cox, 2007a), or optionally in TML or MPEG stream out-of-band.
The optional *transactional* profile is very important whenever the service is not only consumer oriented but also producer oriented: in this case this profile provide an access point for data uploading. The *RegisterSensor* request allows to add to the SOS a new sensor by providing the sensor description (SensorML or TML as responded by a *DescribeSensor* request) and the observation template (O&M) instructing the service on the new observations to be collected and served; in response a sensor unique identifier is returned. The *InsertObservation* facilitate the sensor in registering new observations by providing the previously obtained sensor identifier and the O&M <*observation*> element containing the measured values.
The *enhanced* profiles add other features to the service, like the capability to repeatedly obtain observations with a predefined *GetObservation* template request allowing to save bandwidth (GetResult), or the capability to get the geometry of an advertised feature of interest (*GetFeatureOfInterest*) or the time for witch a feature of interest has been observed (*GetFeatureOfInterestTime*).

2. *SOS INVESTIGATION*

As already stated in the introduction this research aims to investigate the capability of SOS to be applied in a real case for the management and provision of Sensor observed data for the operational flooding forecast system for the Lake Maggiore.

In order to conduct this investigation, the research has been subdivided into three steps:

1. Analysis of existing procedures and needs in the management of the Canton Ticino hydro-meteorological monitoring network.
2. Design of a new SOS software.
3. Analysis of SOS and implementation strategies.

The following sections will describe these three research steps.

2.1 Analysis of IST existing situation, design of future system and identification of needs.

The IST is directly managing a total of 25 station with rain gauge and temperature sensors and 20 with river gauge station and water temperature sensors (see figure 1).

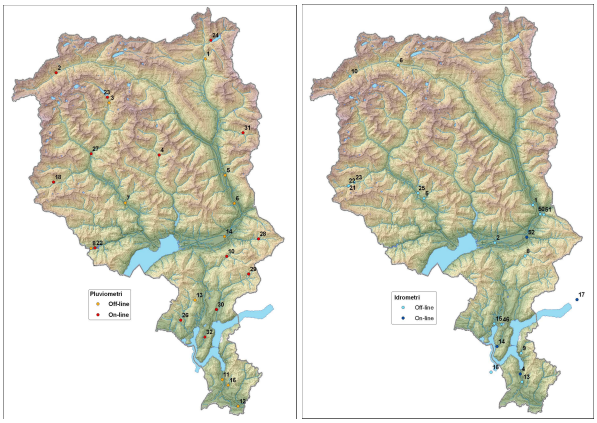

Figure 1 – Canton Ticino hydro-meteorological network.

Up to now, the on-line sensors data were downloaded throughout the GSM line every 6 hours using a vendor proprietary software and transformed to ASCII formatted files. The raw data present in these files, together with the manually collected observations, are then inserted into a database. Moreover, aggregated daily values are generated, verified and inserted into a different database (*historic database*) for official yearly hydrological Canton Ticino bulletin production.
Any data exchange with third parties is performed by using the File Transfer Protocol (FTP) and recorded into a database to be later on used as input of the Lake Maggiore level forecasting model (see figure 2).

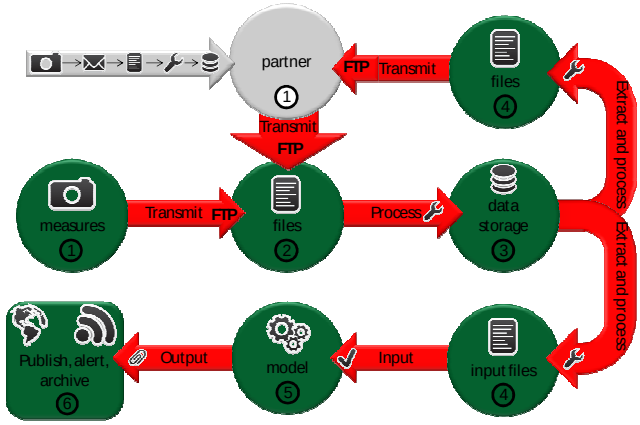

Figure 2 – Current data processing workflow (in light grey third party processes).

Ideally, in the future different parties will use SOS standard and this situation will evolve in the one illustrated in figure 3, where

SOS are used to provide data and Web Processing Service (WPS) is used to provide modelling results.

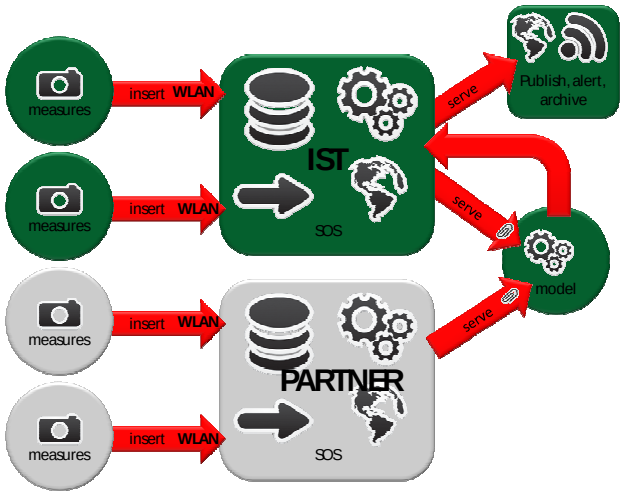

Figure 3 – Future aimed workflow (in light grey third party processes).

Within the conducted work, we focused on the transition of the existing system to the adoption of SOS, leaving for future works the transposition of the hydrological model into WPS.

By migrating the system, we should be able to provide the existing system capabilities, but also some improvements.
The system features that have been individuated together with the hydrologist are listed below:

1. Automatic registration of new measurements from on-line sensors.
2. Manual registration of not on-line sensors.
3. Distribution of observed data.
4. Validation with capability of data correction (either manual or automatic) and assignment of data quality index.
5. Extraction of regular time series with frequency of 10 min., 30 min., 1 h, 3 h, 6 h, 12 h, 1 day, 1 week or 1 month (with option to produce text comma separated values files or plots).
6. Mapping of sensor location and latest measurements.
7. Reporting of sensors properties
8. Conversion of water height data in discharges by using functions that vary in time

2.2 istSOS software design

The choice of developing a new software has been selected because only a few software available on the market support the SOS standard (52° North SOS, MapServer, and Degree SOS) and we wanted to deeply understand the technological issues and new possibilities to satisfy our needs without any link to any existing implementation choice.
istSOS (Cannata and Antonovic, 2010) is a SOS software entirely developed in Python language that is distributed with GPL v.2 licence.
It is based on the *Apache* web server though the *mod_python* package (http://www.modpython.org), it relies on the *PostgreSQL/PostGIS* database for data storage (http://postgis.refractions.net) and take advantage of the *GDAL* library (http://www.gdal.org).
The software has been developed following a *factory* approach that allows the instantiation of specialized class or function specifically built to execute a particular received request.

In summary the software consist in the *sos.py* file that is executed by the Apache server every time a selected URL is accessed. This file performs in sequence the following actions:

1. Load istSOS module and other required libraries.
2. Load the sosConfig file where specific service configuration is set
3. Read the request.
4. Instantiate the sosDatabase object that set and open a connection with the configured istSOS database.
5. Execute the *sosFactoryFilter* function by passing the request and retrieving a filter object.
6. Execute the *sosFactoryResponse* function by passing the filter and the database connection objects and retrieving a response object containing requested information.
7. Execute a *sosFactoryRenderer* function by passing the response object and retrieving the SOS response as XML format.

Each "factory" function is able to understand what specialized method is required, so the *sosFactoryFilter* will call the appropriate filter child object depending on the request type; the *sosFactoryResponse* will call the specialized response function that actually execute the received request (by collecting information or executing transaction); and finally the *sosFactoryRendered* will execute the rendered function able to represent the information in a standard SOS response format.
istSOS, that today is available in its first version 1.0, rely on a specific database model whose SQL creation file is provided together with the software package at the istSOS project Web site (http://istgeo.ist.supsi.ch/software/istsos). The package contains also the istSOS library, easily installable trough the *setuptools* module (http://pypi.python.org/pypi/setuptools), the *sos.py* file and a *sosConfig* file as a template. Other information on the software are available at the istSOS project Web site.

2.3 Individuated issues and implemented strategies

During the development of the software and the migration of the system the authors faced some issues that were mainly related to the usage of O&M standard in the SOS specification.
With reference to the O&M standard, the SOS defines an observation as: "*An act of observing a property or phenomenon, with the goal of producing an estimate of the value of the property. A specialized event whose result is a data value*" and a procedure as: "*Method, algorithm or instrument*".
For sake of simplicity we will use in the following paragraphs the namespaces om, sa and swe respectively for OGC's O&M, Sampling (Cox, 2007b), and SWEcommon (Botts and Robin, 2007) schemes when referencing to XML objects.
In particular, most of the issues encountered during the implementation of the istSOS were related to the *<om:observation>* object (see figure 4) returned in response of a *GetObservation* request.
Each observation element must include the following mandatory XML elements:

- *<om:procedure>*; it defines the process used to obtain the data; in general it provides reference to the name of the advertised procedure.
- *<om:featureOfInterest>*; it defines the target of the measurement, generally expressed as GML and/or reference to advertised feature of interest name.
- *<sa:samplingTime>*; it represent the time at which the result applies to the feature of interest.

- <*om:observedProperty*>; it defines the nature of the observation specifying the semantic type, like "river discharge" or "water temperature". Generally it is expressed as referenced definition.
- <*om:result*>; it provides the place to put the requested data. These, according to the O&M schema can be of any format.

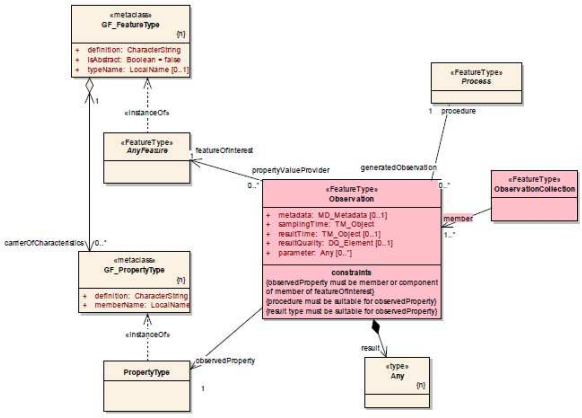

Figure 4 – O&M observation element (source Cox, 2007a)

The first encountered issue is related to the management of irregular time series. It is very important to be able to provide raw data of not regular time series like the ones collected with tipping bucket rain gauges. In this case, observations are present only when there are precipitations and therefore the user need a method to discriminate between missing data due to the absence of rain and due to the absence of monitoring. The implemented solution relies on the value of the <*sa:samplingTime*>; within istSOS it does not refer to the returned observations time lag, (information contained into the time values of returned data), nor to the requested *eventTime* parameter value of the *GetObservation* (already known by the consumer that made the request). istSOS uses this object to provide information on the whole period for which the feature of interest has been observed by that procedure: if a *getObservation* response has no data but the <*sa:samplingTime*> shows that the precipitation where observed in the requested period, the user can infer the information of no rainfall.

The second issue is related to undefined data format. The <*om:result*> in fact accepts any type of children elements to be used for representing the requested data. This fact leads to the practical impossibility of a machine process to automatically interpret all the possible data formats (including users defined formats) and to fuse them with other SOS services or WPS processes.

It is clear, that different types of observations require different types of representation, and that this is dependent on the type of the process (e.g.: rainfall at a given location or temperature of the air along unpredicted paths). For this reason istSOS, defines the process as: *a method, algorithm or instrument that generates a time series and that is identified by the definition of the process type*

Possible process types, are derived by the combination of the following observation properties/characteristics:

In-situ or remote.

Direct or calculated.

Discrete or distributed.

Fixed or mobile.

Currently, the supported process types in the istSOS version 1.1, are: "*in-situ, direct, discrete, fixed*" and "*in-situ, direct, discrete, mobile*" processes. They are represented within the <*om:result*> with the <*swe:DataArray*> element of the Sensor Web Enablement schema (see figure 5).

In the case of fixed sensor the location is identified by the *featureOfInterest*, while in the case of mobile sensor the time varying position is defined by three observed coordinates (east, north and elevation) and the *featureOfInterest* represents the bounding box of all the observations.

```
−<om:result>
 −<swe:DataArray>
  −<swe:elementCount>
   −<swe:Count>
     <swe:value>2</swe:value>
    </swe:Count>
  </swe:elementCount>
  −<swe:elementType name="SimpleDataArray" xlink:href="http://mmisw.org/ont/mmi/obs.owl/timeSeriesDataRecord">
   −<swe:DataRecord definition="http://mmiws.org/ont/x/timeSeries">
    −<swe:field name="Time">
      <swe:Time definition="urn:ogc:def:parameter:x-ist::time:iso8601"/>
     </swe:field>
    −<swe:field name="rainfall">
     −<swe:Quantity definition="urn:ogc:def:property:x-ist::rainfall">
       <swe:uom code="mm"/>
      </swe:Quantity>
     </swe:field>
    </swe:DataRecord>
  </swe:elementType>
  −<swe:encoding>
    <swe:TextBlock tokenSeparator="," blockSeparator="@" decimalSeparator="."/>
  </swe:encoding>
  −<swe:values>
    2010-04-23T11:25:39Z,0.200000@2010-04-23T11:39:02Z,0.200000@2010-04-23T11:54:39Z,0.200000@2010-04-23T12:49:
  </swe:values>
 </swe:DataArray>
</om:result>
```

Figure 5 – example of <*om:result*> implemented by istSOS.

A third issue is related to the *InsertObservation* request. The user has to provide a request with an <*om:observation*> element. By interpreting the SOS standard the authors inferred that this request has been designed for the registration of a single observation; in fact the request name is singular and the response contain the identifier (one) of the registered observation. The schema formally allows to specify a series of observations within the <*om:observation*> element, and therefore the request should provide the capability, as suggested also by the practical operations, to perform multiple value insertion. As implementation choice, the istSOS permits this behaviour in the execution of the request and returns as a response a list of identifier concatenated with the @ symbol. In case of error the service answers with an exception (following the SOS specifications) and no observation is registered.

Moreover, according to the standard there is no option for adjust previously registered observations. This is very important in the management of monitoring data: often the measures need to be post-processed and analysed in order to assign them a quality index and optionally correct the offered data. For this reason istSOS includes an additional non standard parameter that allows to force the insertion; existing observations in the <*sa:samplingTime*> period of the request are deleted and the new observations are inserted.

Other individuated minor factors that, according to the authors, leads to ambiguities but that istSOS have not coped with, are:

- Missing of definition of time series elements.
- Not defined onthologies for observed properties.
- Not properly advertised procedure observation period.

Time series definition should includes the time series type (regular or irregular) and eventually the time series frequency, the aggregation mode, the time of reference, the boundaries (open/close - upper/lower).

In particular the reference time is very important when the user have to deal with aggregated values. In fact in this case the information of 1.2 mm of rain at the 2010-06-14 T12:00 have some degree of ambiguity: is it referring to the period between the 11:00 and the 12:00 or between the 12:00 and the 13:00? Boundaries instants are included or not? What type of aggregation method was applied?

Of course, all these information can be included into the *DescribeSensor* response within a generic element but, in the opinion of the authors, it should be clearly addressed in a mandatory parameter in other to be automatically processed.
Onthologies of observed properties are missing, and this raises ambiguity in interpreting and third party provided observations. For example, hypothetically it could not be clear if an advertised temperature observation refer to water temperature, air temperature at 2 m or at 8 m, or soil temperature. The specification give the freedom to use custom definitions: this add flexibility but at the same time diminish the level of interoperability when a user may need data fusion.
The *GetCapabilities* response advertise for each offering (a group or procedures arbitrarily aggregated): the member procedures, the features of interest, the observed properties and the observed period. But when a user need to know when a given procedure has observations and executes a *DescribeProcess* request, the response not mandatorily contains this information. A better advertisement of procedure and observed period is therefore desirable.

3. CONCLUSION AND FUTURE WORKS

Considering the individuated IST requirements in the migration of the management of the hydro-meteorological network of the Canton Ticino, we can, at this point, assert that SOS, and in particular the istSOS implementation, is capable to fulfil the needs of: automatic registration of new measurements, manual registration of off-line sensors, reporting of sensor properties, and dispatch of raw data.
If istSOS is used, the mapping of sensors location and their latest measurements can be achieved through the development of client-side process that parse the response of a *GetObservation* request with a void time parameter: in this case istSOS answers with the latest observation. Moreover the capability of correct existing information is supported with the new no-standard parameter for forcing the insertion of observations.
What still is missing is the capability to:

- Convert existing water heights into discharges.
- Extract regular time series at different frequencies.

All of this capabilities could be performed by third processes, that most likely would be WPS processes. The drawback of this solution is the lost of performance and of interoperability in a data fusion prospective (where and in what format are data available?).
Currently, at the IST, the authors are working in extending the istSOS capabilities by adding the possibility to define *virtual processes*, that are derived by chaining and/or applying time varying functions to existing procedures.
This will enable the possibility to offer aggregated registered measures as well as derived, but often most useful, properties.
For data reports and plots production, the WPS solution is definitively the most appropriate choice.
Currently SOS is partially open to different interpretation and some aspects are not sufficiently specified or encompassed.
Loosed definitions lead to weak interoperability, and this can be proven by comparing the responses of different SOS software implementation.
Some of the individuated open issues or questions that are encountered at the IST, in the daily management of monitoring data, have been addressed during the development of the new istSOS software. This leads to the development of a new software where specific implementation choices were taken in support of the existing individuated needs. The authors believe that these choices can be considered valid also for other fields of activity, that actually are exploited at the IST, like landslide and water quality monitoring data.
In summary, the authors can overall conclude that the Sensor Observation Service is a very attractive standard for the management and distribution of sensor observed data, and once the existing open questions will be more deeply assessed in the future versions of the standard, will became an essential service for data fusion and service orchestration.

4. REFERENCES

Botts, M. and Robin, A., 2007. OpenGIS® Sensor Model Language (SensorML) Implementation Specification, Version 1.0, OGC document 07-000. *http://www.opengeospatial.org/standards/sensorml* (accessed 07 June 2010)

Botts, M., Percivall, G., Reed, C., Davidson, J., 2007. OGC® Sensor Web Enablement: Overview And High Level Architecture, White Paper, OGC document OGC 07-165. *http://www.opengeospatial.org/projects/groups/sensorweb* (accessed 07 June 2010)

Cannata, M., Graf, A., Salvetti, A., Salvadè, G. (2005), *Sviluppo di un sistema di gestione dei rischi idrogeologici del Lago Maggiore*, ISBN 88-7479-018-x, sections 6.1/6.2, pp. 86-105.

Cannata, M. and Antonovic, M. (2010), istSOS: Sensor Observation Service in Python, *Geomatic Workbooks* n. 9 (ISSN 1591 092X), *http://geomatica.como.polimi.it/workbooks* (in printing).

Cerami, E., St.Laurent, S., 2002. *Web Services Essentials*, O'Reilly & Associates, Inc., Sebastopol, CA, ISBN:0596002246.

Cox, S., 2007a. Observations and Measurements – Part 1 - Observation schema, Version 1.0, OGC document OGC 07-022r1. *http://www.opengeospatial.org/standards/om* (accessed 07 June 2010)

Cox, S., 2007b. Observations and Measurements – Part 2 - Part 2 - Sampling Features, Version 1.0, OGC document OGC 07-022r1. *http://www.opengeospatial.org/standards/om* (accessed 07 June 2010)

Havens, S., 2007. OpenGIS® Transducer Markup Language (TML) Implementation Specification, Version 1.0, OGC document 06-010r6. *http://www.opengeospatial.org/standards/tml* (accessed 07 June 2010)

Na, A. and Priest, M., 2007. OpenGIS® Sensor Observation Service Version 1.0 OGC document 06-009r6. *http://www.opengeospatial.org/standards/sos* (accessed 07 June 2010)