Open Geospatial Consortium, Inc.

Date: 2008-07-1

Reference number of this document: OGC 06-021r2

Version: 0.1.0

Category: Best Practice

Editors: Ingo Simonis

OGC® Sensor Web Enablement Architecture

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.
To obtain additional rights of use, visit http://www.opengeospatial.org/legal/.

Warning

This document defines an OGC Best Practices on a particular technology or approach related to an OGC standard. This document is not an OGC Standard and may not be referred to as an OGC Standard. It is subject to change without notice. However, this document is an official position of the OGC membership on this particular technology topic.

Preface

Suggested additions, changes, and comments on this draft report are welcome and encouraged. Such suggestions may be submitted by email message or by making suggested changes in an edited copy of this document.

The changes made in this document version, relative to the previous version, are tracked by Microsoft Word, and can be viewed if desired. If you choose to submit suggested changes by editing this document, please first accept all the current changes, and then make your suggested changes with change tracking on.

Forward

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium Inc. shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

Contents

Page

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

 Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

Figures

Page

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

Tables

Page

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

OGC® Sensor Web Enablement Architecture

1 Introduction

1.1 Scope

This document describes the architecture implemented by Open Geospatial Consortium's (OGC) Sensor Web Enablement Initiative (SWE). In contrast to other OGC SWE standards, this document is not an implementation standard.

In much the same way that HTML and HTTP standards enabled the exchange of any type of information on the Web, the SWE initiative is focused on developing standards to enable the discovery of sensors and corresponding observations, exchange, and processing of sensor observations, as well as the tasking of sensors and sensor systems. The functionality that OCG has targeted within the Sensor Web includes:

- Discovery of sensor systems, observations, and observation processes that meet our immediate needs
- Determination of a sensor's capabilities and quality of measurements
- Access to sensor parameters that automatically allow software to process and geolocate observations
- Retrieval of real-time or time-series observations and coverages in standard encodings
- Tasking of sensors to acquire observations of interest
- Subscription to and publishing of alerts to be issued by sensors or sensor services based upon certain criteria

Within the SWE initiative, the enablement of such a Sensor Web is being pursued through the establishment of several encodings for describing sensors and sensor observations, and through several standard interface definitions for web services. Sensor Web Enablement standards that have been built and prototyped by members of the OGC include the following OpenGIS Specifications:

1. **Sensor Model Language (SensorML)** – standard models and XML Schema for describing the processes within sensor and observation processing systems; includes common data representation models valid for all SWE encodings and service interface standards; provides information needed for discovery, georeferencing, and processing of observations, as well as tasking sensors and simulations. [OGC 07-000 & OGC 07-122r2 (corrigendum)]

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

2. **Observations & Measurements (O&M)** - The general models and XML encodings for observations and measurements made using sensors. [OGC 07-002r2 & OGC 07-022]

3. **Transducer Model Language (TML)** – Conceptual approach and XML encoding for supporting real-time streaming observations and tasking commands from and to sensor systems. [OGC 06-010r6]

4. **Sensor Observation Service (SOS)** – An open interface for a service by which a client can obtain observations and sensor and platform descriptions from one or more sensors. [OGC 06-009r6]

5. **Sensor Planning Service (SPS)** – An open interface for a service by which a client can 1) determine the feasibility of collecting data from one or more sensors or models and 2) submit collection requests to these sensors and configurable processes. [OGC 07-014r3]

6. **Sensor Alert Service (SAS)** – An open interface for a web service for publishing of and subscribing to deliverable alerts from sensor or simulation systems. [OGC 06-028r5]

7. **Web Notification Service (WNS)** – An open interface for a service by which a client may conduct asynchronous dialogues (message interchanges) with one or more other services. [OGC 06-095r1]

The sensor web standards infrastructure defined by these specifications constitutes a revolution in the discovery, assessment and control of live data sources and archived sensor data. The goal of this document is to discuss design and operational concepts for the SWE Architecture.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The OGC shall not be held responsible for identifying any or all such patent rights.

1.2 Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

Contact	Company	Address	Phone	Email
Ingo Simonis (Editor)	Geosptial Research, Germany	Margarete-Bieber-Weg 11 35396 Giessen Germany	+49 641 399 3821	ingo.simonis@geospatialresearch.de
Mike Botts	University of Alabama in Huntsville	ESSC / NSSTC Huntsville, AL 35899	+01-256-961-7760	mike.botts@uah.edu
Alexandre Robin	Spot Image		+33 562 19 4362	Alexandre.Robin@spotimage.fr
John Davidson	ImageMatters Inc., USA		+1 (703) 669-5510	johnd@imagemattersllc.com

 Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

Simon Cox	CSIRO Exploration and Mining	ARRC PO Box 1130 Bentley WA 6102 Australia	+61 8 6436 8639	Simon.Cox@csiro.au
Thomas Usländer	Fraunhofer IITB	Fraunhoferstr. 1 76131 Karlsruhe, Germany	+49 721 609 1480	thomas.uslaender@iitb.fhg.de

1.3 Revision history

Date	Release	Editor	Primary clauses modified	Description
2006-01-20	0.0	Botts/Simonis	throughout	Initial version
2006-02-13	R1	Botts/Robin	throughout	Completed draft
2008-03-31	R2	Simonis	throughout	Reflects all changes/updates until March 2008
2008-07-01		Carl Reed	Various	Preparation for publication as OGC BP

1.4 Future work

This document reflects the current situation of SWE. When SWE specifications will evolve, it is desirable that this document will be updated accordingly.

2 References

The following documents are referenced in this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

OGC 06-121r3, *OpenGIS® Web Services Common Specification*

NOTE This OWS Common Specification contains a list of normative references that are also applicable to this Implementation Specification.

OGC 07-000, *OpenGIS® Sensor Model Language (SensorML) Implementation Specification*

OGC 07-122r2, *OpenGIS® Corrigendum to OGC 07-000, Sensor Model Language (SensorML) Implementation Specification*

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

OGC 07-022, *OpenGIS® Observation & Measurement – Part 1 – Observation Schema*

OGC 07-002r2, *OpenGIS® Observation & Measurement – Part 2 – Sampling Features*

OGC 06-010r6, *OpenGIS® Transducer Model Language (TML) Implementation Specification*

OGC 06-009r6, *OpenGIS® Sensor Observation Service Implementation Specification*

OGC 07-014r3, *OpenGIS® Sensor Planning Service Implementation Specification*

OGC 06-028r5, *OpenGIS® Sensor Alert Service Implementation Specification*

OGC 06-095r1, *OpenGIS® Web Notification Service Implementation Specification*

SANY (2007): Specification of the Sensor Service Architecture

3 Terms and definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Specification [OGC 06-121r3] and in OpenGIS® Abstract Specification Topic TBD: TBD shall apply. In addition, the following terms and definitions apply.

3.1
Sensor
text of the definition

3.2
Sensor System
text of the definition

3.3
Sensor Network
text

3.4
Sensor Web
Text

3.5
Transducer
Text

3.6
Actuator
Text

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

4 Conventions

4.1 Abbreviated terms

API	Application Program Interface
COTS	Commercial Off The Shelf
DCE	Distributed Computing Environment
O&M	Observation and Measurement
SAS	Sensor Alert Service
SensorML	Sensor Model Language
SOS	Sensor Observation Service
SPS	Sensor Planning Service
SWE	Sensor Web Enablement
TML	Transducer Model Language
WNS	Web Notification Service

4.2 UML notation

Some diagrams that appear in this standard are presented using the Unified Modeling Language (UML) static structure diagram, as described in Subclause 5.2 of [OGC 06-121r3].

5 Sensor Web Overview

5.1 Sensor Web

The Sensor Web is a revolutionary concept towards achieving a collaborative, coherent, consistent, and consolidated sensor data collection, fusion and distribution system. It can be viewed as a new breed of Internet for monitoring spatio-temporal phenomena appearing in the physical environment in real time. Any kind of sensor, from a thermometer located at a fixed position to a complex hyper-spectral sensor on board of an earth-orbiting satellite, will be made available on a global level in the near future. Sensors remain at fixed locations (e.g. as part of weather stations) or move autonomously or remotely controlled in physical space (e.g. on board of vehicles, airplanes or satellites). Once deployed, each sensor associates the phenomenon it senses with the location it currently populates. This information is either stored on the sensor for later access or directly sent to aggregation systems. The retrieval and processing of sensor data, but also the management of sensor devices (i.e. tasking), will soon be carried out by means of distributed software entities that interoperate via the Internet. At its stage of completions, millions of sensors will be connected to a large network and produce georeferenced observa-

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

tion data. Every single sensor provides a small mosaic stone that helps us to generate a consolidated view of the world, to get a better understanding of the past, present, and future situation of our planet as well as active processes and correlations.

The Sensor Web represents a meta-platform that integrates arbitrary sensors and sensor networks; each maintained and operated by individual institutions, e.g. the Australian Water Resources Network, the European Environment Information and Observation Network, or the South African Earth Observation Network. This reflects the existing legal, organizational and technical situation. Sensors and sensor systems are operated by various organizations with varying access constraints, security, and data quality and performance requirements. The architectural design of the Sensor Web allows the integration of individual sensors as much as the integration of complete sensor systems without the need of fundamental changes to the legacy systems.

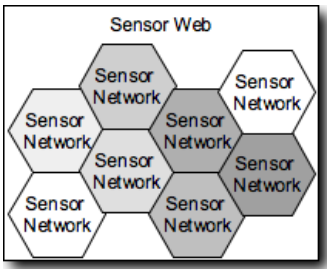


Figure 5-1: Sensor Web: Aggregation of Sensor Networks

Once connected to the Sensor Web, data sets may get used multiple times in applications never intended by the original system setup. Traffic sensors that have been deployed initially to avoid jams by dynamic traffic control might get used to calculate the carbon dioxide ratios of highway sections in another application. Satellites with different sensors on board might get used in a variety of application domains that were not primarily targeted, simply due to interoperable interfaces that allow users to task the satellite based on distinct requirements [4].

5.2 Sensor Model

5.2.1 Overview

The OGC Sensor Model is best described using a number of different views [1]. We will use the five viewpoints defined in the ISO RM-ODP approach to shed light on the various aspects of sensors. The following discussion starts with the Technology Viewpoint, illustrating the view of a hardware manufacturer, and then reflects a "Sensor" from the Engineering, Service and Informational Viewpoint. The Enterprise Viewpoint is not considered in this discussion.

> *Note: In this discussion, the thing observed by sensors is called "observed property" in line with the OGC Observations and Measurements model (OGC 07-022). An observed property identifies or describes the phenomenon for which the observation result provides an estimate of its value. Based on this definition, we define a sensor to be an entity that provides information about an observed property at its output. A*

 Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

sensor uses a combination of physical, chemical or biological means in order to estimate the underlying observed property. At the end of the measuring chain electronic devices produce signals to be processed.

5.2.2 Technology Viewpoint

From a technical point of view, we consider a sensor to be a device that responds to a (physical) stimulus in a distinctive manner, e.g. by producing a signal. This means that a sensor device converts the stimulus into an analogue or digital representation, the latter being of more interest within the IT domain. In contrast, an "actuator" transforms a signal into an action that has some sort of effect on the physical domain, i.e. the actuator produces a stimulus that can be observed by a sensor. Figure 5-2 illustrates this definition.

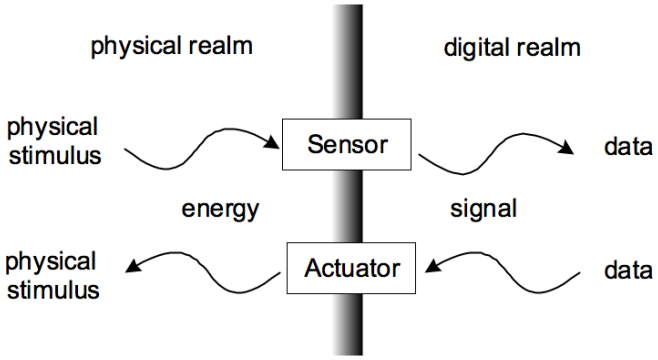


Figure 5-2: Sensor and actuator model (derived from (Ricker/Havens, 2005))

The following sections provide a more detailed discussion on sensors and distinguish between simple and complex forms of sensors and sensor systems.

5.2.2.1 Simple Form of a Sensor

The sensor observes an observed property which may be a biological, chemical or physical property in the environment of a sensor, at a specific point in time (t0) at a specific location (spRef), i.e. within a temporal and spatial context. Note that the location of the sensor might be different from the location of the observed property. This is the case for all remote-observing sensors, e.g. cameras, radar, etc. For an in-situ observing sensor, locations of sensor and observed property are identical. The simple form of a sensor provides information on single observed property. Figure 5-3 shows the model of this situation.

The observed property is usually converted to a different internal representation, usually electrical or mechanical, by the sensor. Any internal representation of the observed property is called a signal. Within the sensor any kind of signal processing may take place. Signal processing typically includes linearization, calculations based on calibration coefficients, conversions to different representations and any calculations to prepare the sensor data for output. The signal may also be transferred over longer distances.

Note: This transfer is not restricted to a signal transmission over a communication network but could also be a human carrying a chemical probe (e.g. a water probe from a river) to a laboratory.

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

The path from signal observation to the output of the signal processing takes time and may also be distributed to several locations. However, the temporal context (t0) and the spatial context (spRef) of the signal observation must be preserved. As an example, consider the water probe mentioned above: It is imperative to preserve the time and the location at which the probe has been taken. The time and location of the examination of the chemical probe in the laboratory is only additional meta-information whose relevance depends on the application context.

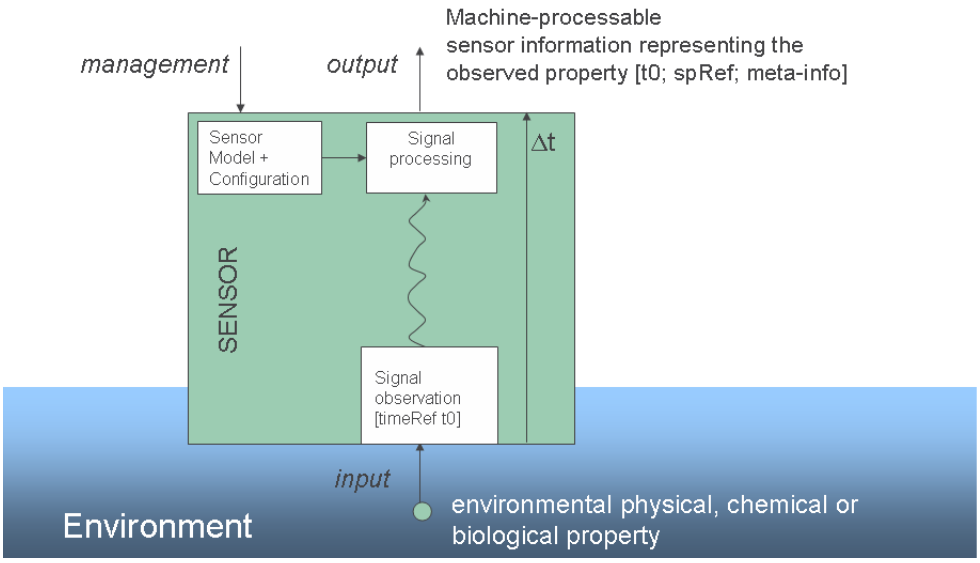


Figure 5-3: Model of a simple form of a Sensor

Finally, the observed property is accessible at the output of the sensor in a machine processable representation. The output provides information about the time (t0) and spatial context (spRef) during observation, though those parameters are usually provided in the form of meta-information and not as part of the observation result. Due to the delay ît produced by the sensor during the observation, the information at the output of the sensor cannot be accessed before t0+ît. This ît can take any range from microseconds to several weeks or months.

Different sensors may provide different representations of the same observed property. They may differ in the units used, the quality of the representation, the observation method used or the internal signal processing. The value that represents the observed property may be a single value, a range of values, a choice between the worst and best value, a sequence of values or a multi-dimensional array of values representing, for example, a picture. It may contain values for each point in spatial/temporal context or may be a statistical representation in space or time. The description of the representation as well as all other observation related information has to be provided as sensor meta-information at the sensor output to be used by an application. A sensor may internally store representations of an older temporal context (history) or spatial context.

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

In addition to its output, a sensor may provide an interface to perform the management of the sensor itself. For instance, this interface may be used to tag the sensor with a name, to configure the internal signal processing or to monitor the behaviour of a device.

5.2.2.2 Complex form of a Sensor

If an observed property cannot be observed with available sensor technology of simple form, it is possible to build a complex form of a sensor using several simple forms. This composite model is illustrated in Figure 5-4.

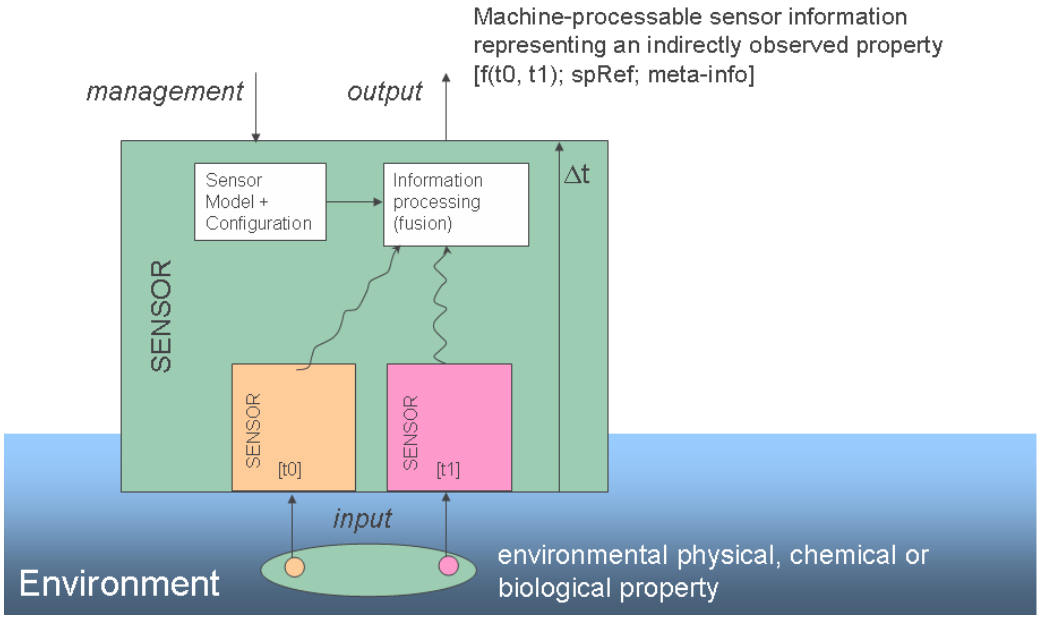


Figure 5-4: Model of a complex form of a Sensor

The information about the observed properties of the individual components of the complex form may be processed by any method of information processing (e.g. in fusion blocks). The output of the complex form of a sensor represents an observed property as defined by the sensor operator. This means that the linkage of the output of the complex form of a sensor to the output to the simple forms of a sensor is transparent. Still, even the complex form has to provide some information about the temporal and spatial context of its output data.

> *Note: Those contexts might be of different scale: A complex form of a sensor might provide forecast information for the next multi-week period in a large area, whereas the simple forms provided observations at single points in time and space only.*

Thus, the resulting temporal context of complex forms of sensors is a function of the temporal contexts of the individual observed properties, represented in Figure 5 5 as f(t0, t1). The same may be true for the spatial context, in Figure 5 5 represented as spRef = f(spRef1, spRef2). The function should be provided as part of the meta-information, in-

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

cluding information about all processing steps at the output interface as well as on the management interface.

Depending on the application context, this complex form of a sensor may itself be aggregated into another complex form. In this case, the internal structure is a black box to the application.

5.2.2.3 Sensor System

Several sensors may be combined within a sensor system (see Figure 5-5) that allows the management of the system holding the sensors in addition to the management of each individual sensor separately. This is done through the management interface of the sensor system.

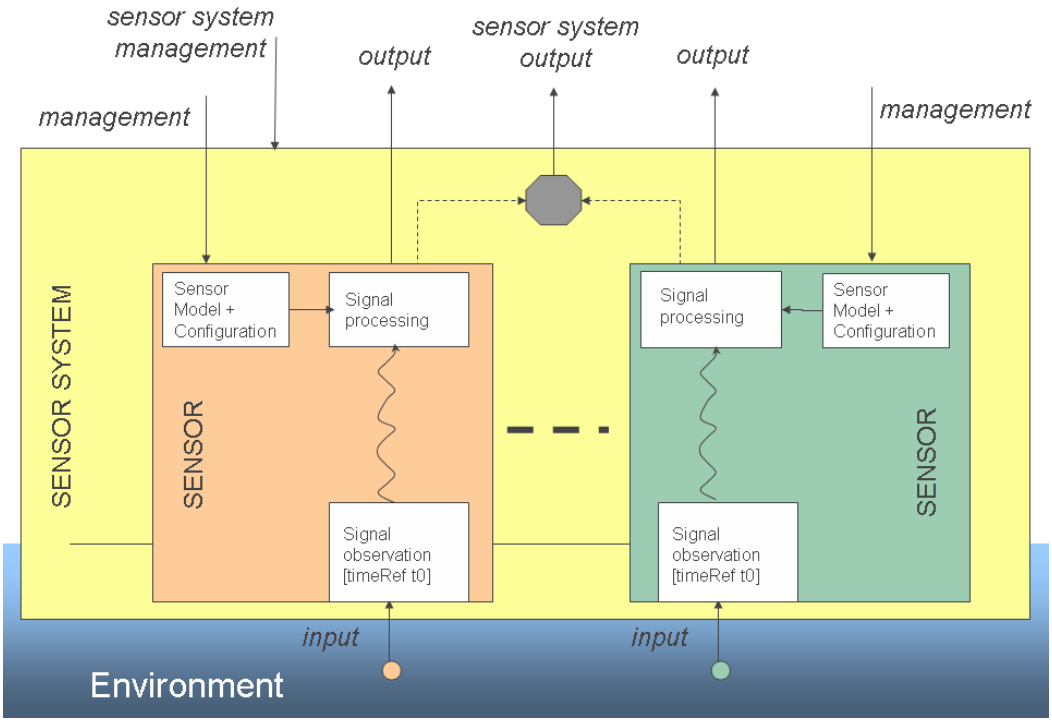


Figure 5-5: Model of a Sensor System

Note: The important characteristic of a sensor system is its spatial homogeneity and its singular output and management interfaces that reflect its organisational unit. Spatial homogeneity here means that the sensor system integrates any number of sensors into a single spatial reference, i.e. a geometry object such as a point, a polygon or even a three-dimensional body, and makes them available at a joint interface (individual sensors might be still accessible via separate interfaces). Examples for sensor systems are satellites (whereas the physical structure of the satellite is a platform, not a sensor) with a number of remote-observing devices, weather stations with sensors for wind speed, temperature, and humidity, ground water observation systems used for surveillance of the environment around a chemical plant or a system of sur-

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

face water observation points ordered on the surface and in the depth of a water body. In this way, a sensor system might be a purely logical unit representing data for an area based on point-based observations.

5.2.3 Engineering Viewpoint

Roughly speaking, the engineering viewpoint links components to a communication network. The network might be the Internet or any other open communication network. The components themselves implement purposes, functions, and content as described in the service and information viewpoint below. Thus, the sensor model is extended with a network node component (e.g. an Internet node) as illustrated in Figure 5-6.

Sensor systems might be either connecting a single sensor (a) or a whole sensor network (c) to the communication network. Further on, a sensor system might even integrate all necessary components to act as one single network node, i.e. the sensor system is addressable and accessible within the communication network (b).

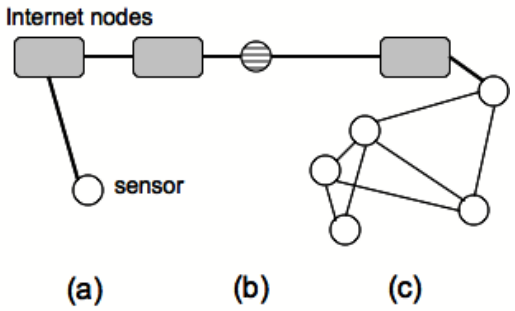


Figure 5-6: Sensors connected to a Communication Network (here: Internet node)

5.2.4 Computational Viewpoint[1]

The service viewpoint is concerned with the functional decomposition of the system into a set of services that interact at interfaces. Transferring this software modelling perspective into a more functional system perspective of our sensor model brings us to an even more complex view of a sensor system. There are two perspectives for the service viewpoint: an internal perspective and an external perspective.

The internal perspective ignores the communication part for a moment and has a closer look at the physical device called sensor by converting the black box sensor into a white box, (see Figure 5-7).

The sensor responds to the physical stimulus “temperature” with the generation of a certain voltage observed in Volts. Afterwards, the temperature gets converted in its digital representation, observed in Kelvin. As the stimulus “temperature” gets converted into its

[1] Sometimes referred to as *Service Viewpoint*

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

digital representation “temperature”, we would still consider the device a sensor, even though the term sensor system becomes more appropriate. In this case, we use a direct conversion of the observed voltage to a temperature value defined in Kelvin.

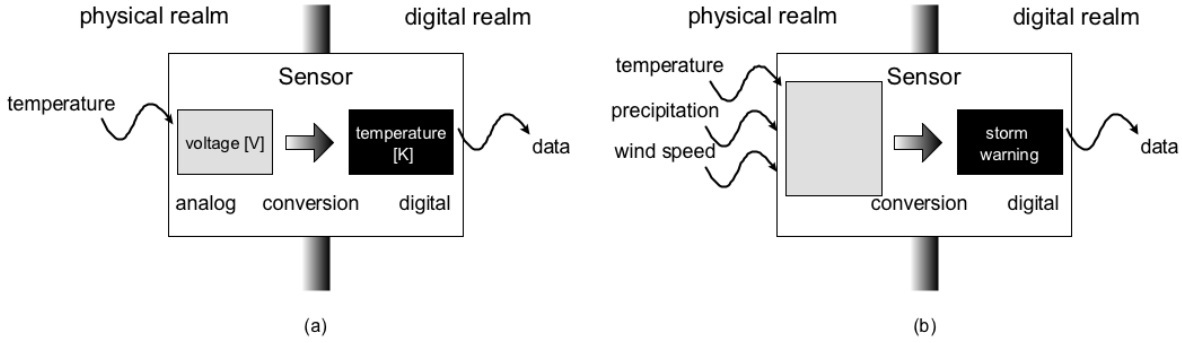


Figure 5-7: Sensor Models, white box

The external perspective represents the view of a software developer or a designer that aims at integrating a sensor into a network of services. From this perspective, a sensor may be seen as a component in a service network with two major interfaces:

- one interface to access the data that represent the property observed by the sensor, and

- a management interface that allows one to configure and monitor the internal behaviour of the sensor (see the internal perspective).

Both interfaces have been illustrated before in Figure 5-3, Figure 5-4, and Figure 5-5. An example of a sensor data access interface is the OGC Sensor Observation Service as described in section 7.5. An example of a management interface to a sensor is the OGC Sensor Planning Service as described in section 7.7.

From the service viewpoint, it often makes sense to consider a simulation model as a sensor, because a model can provide data for times in the past or future analogous to a sensor device. This view is, for example, found in [3]. The main reason for this very broad usage of the term “sensor” results from research and standardization efforts within the domain of service-oriented architectures. Here, web services providing access to sensor data are often called 'Sensor Services'. As long as sufficient meta-information comes along with the data (e.g. how the data were produced, quality etc.), it does not make any difference for the client if physical devices or simulation models produced the data. This approach has the advantage that generic sensor applications may be built that retrieve their data from physical sensors (usually past observation results) in the same way as from simulation models (i.e. calculated future observation results in case of prediction models).

5.2.5 Information Viewpoint

The information viewpoint is concerned with the semantics of information and information processing. Thus, it discusses a sensor in regard to the semantics behind a sensor or

 Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

sensor system. The abstraction from the physical device described in the technology viewpoint becomes appropriate.

We speak of a sensor as a source that produces a value within a well-defined value space of an observed property which may represent a physical, biological or chemical environmental phenomenon. Sensors and sensor systems as well as simulation models fulfill this definition. If the semantics did not differentiate between produced data based on a physical stimulus or any other data, the limit between model and sensor disappears.

In the information viewpoint, we abstract from the source of observation data and concentrate on the data that are provided in the form of observation results. Those results have to follow the sensor data information model, i.e. the results have to reflect all aspects of the underlying viewpoints. In addition to the observation result, information about the observation procedure, spatial-temporal context, and organizational characteristics have to be provided. Such information is considered to be meta-information for the purpose of interpretation and further processing of the observation results.

6 Sensor Web Enablement

Sensor Web Enablement (SWE) extends the OGC web services and encodings framework by providing additional models, services and encodings to enable the creation of web-accessible sensor assets through common interfaces and encodings. SWE services are designed to enable discovery of sensor assets and capabilities, access to those resources and data retrieval, as well as subscription to alerts, and tasking of sensors to control observations. Though the term “sensor” is used expressis verbis, sensor assets may includes observation archives, simulations, and observation processing algorithms in addition to physical sensors. SWE enables interoperability between disparate sensors, simulation models, and decision support systems. It acts as a middleware layer between physical assets and automated tools or tools operated by humans, as illustrated in the figure below.

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

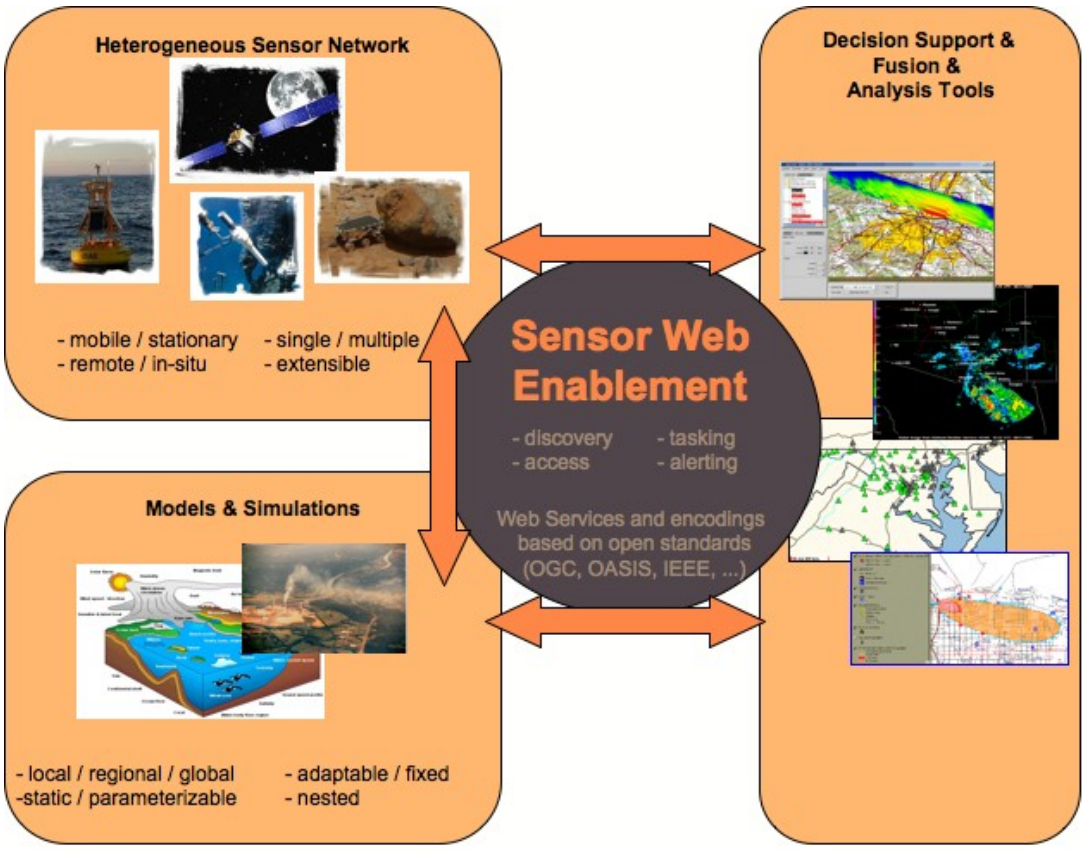


Figure 6-1: SWE Framework

As an advanced architecture, SWE covers all steps of the operation cycle illustrated in the following figure below.

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

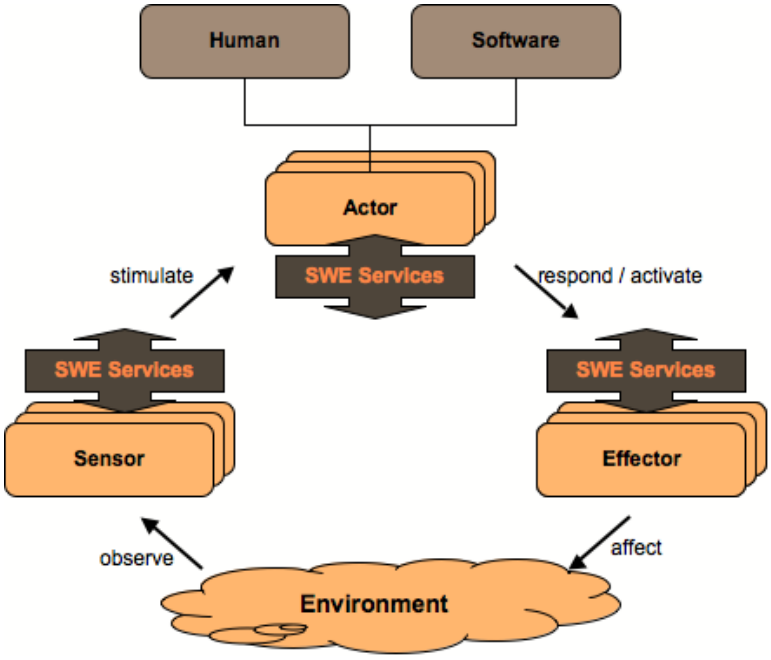


Figure 6-2: SWE Operation Cycle

Humans might be part of this cycle, which otherwise operates fully automatically. SWE services and encodings support the observation of the environment, decision-making, and influencing the environment.

6.1 Motivation

It has long been recognized that the current state of sensor networks is that they are developed around different communities of sensor types and user types, with each community typically relying on its own stovepipe system for discovery, accessing observations, receiving alerts, and tasking sensor systems and models. Even within fairly coherent communities, each type of sensor tends to be accompanied by its on metadata semantics, its own data formats, and its own software.

Within such stovepipe systems, the ability to discover and utilize a new sensor asset is typically hindered by incompatible encodings and services. Additionally, readily available information regarding the sensor system, the observation encodings, processing, and supporting services is typically lacking, scattered, or incomplete. Within these systems, adding support for a new sensor asset to an existing decision support tool or processing operation takes at best several days, and at worst many months or years, accompanied by high expense.

As described in the Introduction, the Sensor Web Enablement (SWE) framework has been designed to enable solutions that meet the following desires:

- **Discovery of sensors, observations, and processes** – we wish to enable one to easily discover all sensor assets (sensor systems, simulations, and data processes) that are available for meeting that individuals needs in a timely fashion; this is

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

particularly important, for example, during rescue or mitigation operations following an unexpected disaster or attack

- **Determination of a sensor's capabilities and an observation's reliability** – we wish to provide the ability to readily assess the capabilities of a sensor or simulation system, as well as provide sufficient lineage of an observation to determine its reliability for decision support
- **Access to parameters and processes that allow on-demand processing of observations** – we wish to provide the means to sufficiently be able to support on-demand geolocation and processing of sensor observations by generic software, without the need for a priori knowledge of the sensor system
- **Retrieval of real-time or time-series observations in standard encodings** – we wish to be able to access and immediately utilize observations from newly discovered sensors within decision support tools, models, and simulations without needing to develop sensor-specific readers
- **Tasking of sensors and simulators to acquire observations of interest** – we wish to be able to task a sensor or simulation system, and to provide my collection requirements, using a common interface; this interface needs to be able to support tasking as simple as controlling a web cam, as well as something as sophisticated as a military surveillance operation
- **Subscription to and publishing of alerts based on sensor or simulation observations** - we wish to provide a means by which a sensor system or simulation can publish possible alerts to be issued by sensors or sensor services based upon certain criteria, and allow one to subscribe to and receive these alerts when criteria are met; such criteria could be a simple as a measured value exceeding a certain threshold or as complex as pattern recognition within a single or multiple observations

6.2 Approach

Within the SWE initiative, the enablement of such sensor webs is being pursued through the establishment of three encodings for describing sensors and sensor observations, and through four standard interface definitions for web services:

Sensor Model Language (SensorML) – standard models and XML Schema for describing sensors systems and processes; provides information needed for discovery of sensors, location of sensor observations, processing of low-level sensor observations, and listing taskable properties

Observations and Measurements Schema (O&M) – standard models and XML Schema for encoding observations and measurements from a sensor, both archived and real-time

Transducer Model Language (TransducerML) – Conceptual approach and XML Schema for supporting real-time streaming of data to and from sensor systems

 Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

Sensor Observations Service (SOS) – standard web service interface for requesting, filtering, and retrieving observations and sensor system information

Sensor Planning Service (SPS) – standard web service interface for requesting user-driven acquisitions and observations, to (re-)calibrate a sensor or to task a sensor network

Sensor Alert Service (SAS) – standard web service interface for publishing and subscribing to alerts from sensors

Web Notification Services (WNS) – standard web service interface for asynchronous delivery of messages or alerts from any other web service

6.3 SWE Services and Encodings Interaction

The SWE Services and Encodings interactions are illustrated in the following figures. In the upper right corner, Figure 6-3 shows sensors that are registered at a SOS and publish observation results to the services itself (then we speak of a Transactional-SOS) or into a database which will be accessed by the SOS service instance. To be discoverable, sensors, using SensorML, and SOS register at a catalog service. The user in the lower left corner requires observation data and sends therefore a *search request* to the catalog. The catalog responses with a list of SOS service instances that fulfill the requirements. Eventually, the user binds the SOS and retrieves the observation data, encoded in O&M.

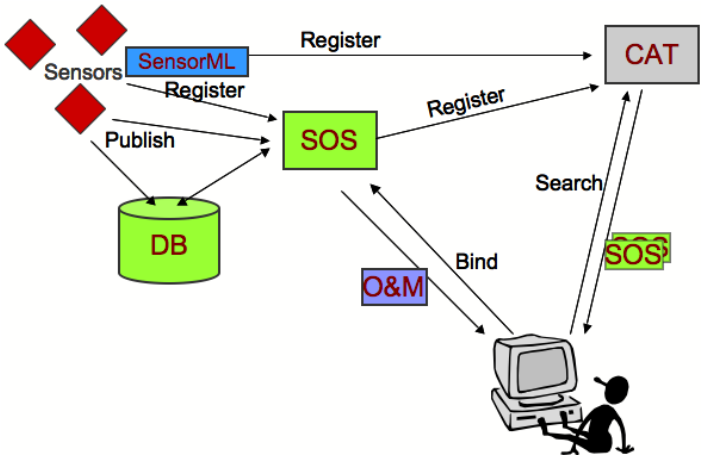


Figure 6-3: SWE Services and Encodings Interactions, part 1

The situation gets a bit more complex in the following Figure 6-4. Let's assume that the catalog didn't provide any SOS instances that fulfill the requirements set by the user. In this case, the user may search for Sensor Planning Services that could task sensors to perform appropriate actions in order to produce the observation data our user is looking for. The catalog provides the link to the SPS instance and the user assigns the task. The SPS forwards the command to the sensor. The communication between the SPS and the sensor

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

is opaque for the user. If we imagine a satellite that has to reorient its infrared cameras and has to reach its target position in space, the tasking might take a while. Once observing the requested scene, the sensor dumps its data into a database that is linked to a SOS. The SPS informs the user about data availability using a Web Notification Service. This has the advantage that the SPS can respond to the tasking request right away and has a mechanism to reach the user at a later stage, e.g. if the data is available or if the tasking is delayed or cancelled. The notification message contains all necessary information to access the data from a SOS.

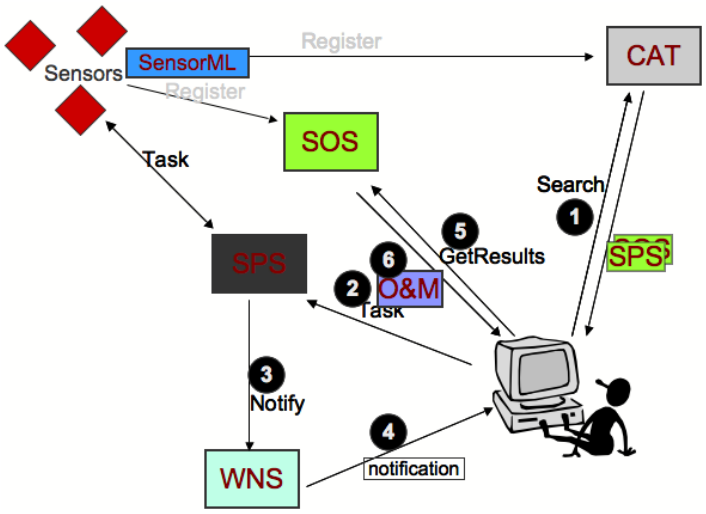


Figure 6-4: SWE Services and Encodings interactions, part 2

Often, we are faced with the situation that a client is not interested in all observation results of a sensor, but wants to get notified immediately if a specific situation is observed. Figure 6-5 illustrates this scenario.

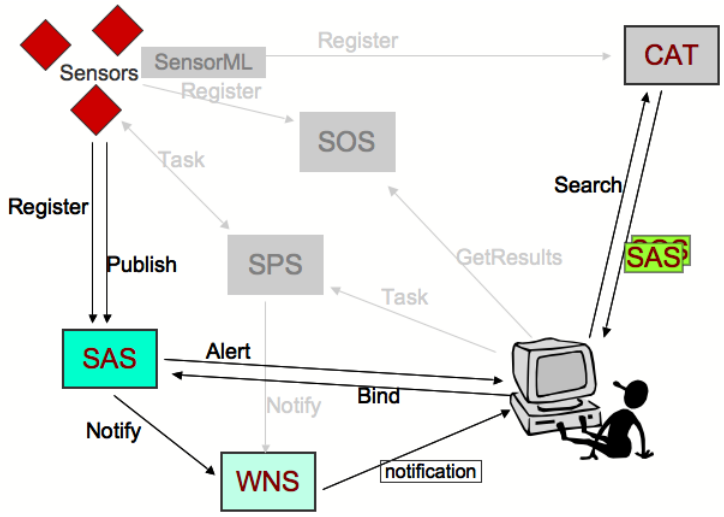


Figure 6-5: SWE Services and Encodings interactions, part 3

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

Once again, the client receives information about appropriate SAS from a catalog and subscribes to the SAS. Sensors publish observation results continuously to the SAS. The SAS handles all the filtering and alerts the client if the subscription condition is matched. The SAS either sends the alert directly to the client, or makes use of the WNS in order to deliver the alert message.

6.4 Example Scenarios

As an example of a use-case scenario in which a SWE-enabled sensor web capabilities might support, consider the need for remedial action following a massive explosion and the subsequent widespread burning of toxic materials. A small sampling of the interactions with web-enabled sensor components could include:

1. an emergency management team (e.g. FEMA) uses sensor registry to discover that NASA has an airborne sensor capable of providing thermal imaging through smoke and clouds (note: there was an actual need to discover such a sensor following the WTC attacks, but the capabilities of and access to the sensor were not discovered until many days after the real need passed) … team places a request for acquisition through SPS and within the hour receives notification through WNS that task is approved and scheduled

2. wind profilers in surrounding area are discovered through sensor registries… measurements obtained (through SOS), geolocated (using SensorML), and applied to simulations of aerosol plume transport … results predict the downwind dispersion location of toxic cloud over the next 6-12 hours

3. biochemical “sniffers” are discovered that are available through secure connections as part of the homeland security initiative .... sensor capabilities are queried and it is determined (using SensorML) that these sensors are capable of detecting the chemical species in question and have high enough sensitivities to detect probable concentrations … team subscribes to alerts when concentrations above a specific value are detected (notice to be sent by email/SMS (Short Message Service) to FEMA personnel and by URL to plume simulation service for automatic updating and refinement of its model runs)

4. through a sensor registry, mobile sensors for monitoring toxic aerosols are discovered to be available through a regional commercial emergency response team … monitor deployment requested and authorized through SPS … FEMA subscribes to alerts from these sensors when certain concentration thresholds are exceeded … alerts sent automatically (by SAS) from field sensors via wireless network along with locations and measurements … containment and medical teams sent to appropriate areas to tend to the local situation

5. in meantime, thermal imager observations taken … notifications sent (via WNS) to fire and hazards teams … unprocessed imagery (streaming as TML data) is obtained through SOS, as are process descriptions (in SensorML) that allow on-demand geolocation and processing of thermal data … location of main hot-spots and source of main toxic located … fire containment is focused on these areas

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

6. sensors onboard emergence response vehicles monitor not only the location of these vehicles at all times, but some also provide mobile measurement of airborne toxin concentrations; real-time video streams (TML) from sensors onboard UAVs and other airborne platforms are provided through SOS services along with SensorML processes for geolocation of video, allowing real-time hazard assessment and rescue monitoring

7. all of these data and services are available to one or more SWE-enabled decision support tools that are capable of processing and fusing this information without *a priori* knowledge of any of the sensor systems involved

This is, of course, only a single possible scenario of the application of SWE supported assets. Additional scenarios have been created for autonomous sensor webs where sensor systems are capable of subscribing to alerts from other sensors and modifying their own sensing behavior based on these alerts.

7 Implementation components

The SWE components serve different purposes with one aspect being common to all components: The usage of fundamental types for data and metadata modeling and encoding. Those aspects common to all SWE standards are defined in a separate namespace called SWE Common. We will briefly describe SWE Common before an overview of the SWE components listed in Section 6.2 is given. Detailed information is provided within separate documents, one for each encoding or service.

7.1 SWE Common

There are several common core definitions used throughout the SWE framework that have been pulled from other SWE specifications, such as O&M and SensorML, and have been placed within the SWE Common namespace. These are currently not defined within a separate document, but rather are defined within SensorML (mostly) or O&M specification documents. Future releases will separate SWE Common definitions into a separate document. Discussion in this direction just have started (end of 2007).

SWE Common knows a number of fundamental types that derive from the *AbstractDataComponent* (which is derived from *gml:AbstractGMLType*, i.e. it implements abstract-types of the *Geography Markup Language, GML*). The scalar data component types use *Range* elements to define minimum and maximum.

- *Count* represents a ‘countable’ property that can be quantified using an integer value. A *CountRange* provides minimum and maximum values as an integer pair in that order.
- *Quantity* represents a numerical that can be quantified using a decimal value. A *QuantityRange* provides minimum and maximum values as a decimal pair in that order.

 Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

- *Boolean* is a simple data component that represents a property that can be true or false
- *Text* is a simple data component that represents a property that takes a string as its value
- *Category* represents textual data that is a member of some larger grouping of values
- *Time* is supported and treated as a special type of numerical scalar. A *TimeRange* provides minimum and maximum values as a time value pair in that order.

These basic data types can be grouped within any of several aggregate objects. Those aggregate objects are described in detail in the SensorML specification. SWE Common provides a number of different encoding options also.

7.2 Sensor Model Language (SensorML)

The measurement of phenomena that results in an observation consists of a series of processes, beginning with the processes of sampling and detecting and followed perhaps by processes of data manipulation. The division between measurement and "post-processing" has become blurred with the introduction of more complex and intelligent sensors, as well as the application of more on-board processing of observations. The typical Global Positioning System (GPS) sensor is a prime example of a device that consists of basic detectors complemented by a series of complex processes that result in the observations of position, heading, and velocity.

SensorML defines models and XML Schema for describing any process, including measurement by a sensor system, as well as post-measurement processing. SensorML supports a variety of needs within the sensor community, including:

Discovery of sensor, sensor systems, and processes - SensorML is a means by which sensor systems or processes can make themselves known and discoverable. SensorML provides a rich collection of metadata that can be mined and used for discovery of sensor systems and observation processes.

On-demand processing of Observations - Process chains for geolocation or higher-level processing of observations can be described in SensorML, discovered and distributed over the web, and executed on-demand without a priori knowledge of the sensor or processor characteristics.

Lineage of Observations - SensorML can provide a complete and unambiguous description of the lineage of an observation. In other words, it can describe in detail the process by which an observation came to be, covering the entire life span from acquisition by one or more detectors to processing and perhaps even interpretation by an analyst. Not only can this provide a confidence level with regard to an observation, in most cases, part or all of the process could be repeated, perhaps with some modifications to the process or by simulating the observation with a known signature source.

Support for tasking, observation, and alert services - SensorML descriptions of sensor systems or simulations can be mined in support of establishing OGC Sensor

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

Observation Services (SOS), Sensor Planning Services (SPS), and Sensor Alert Services (SAS). SensorML defines and builds on common data definitions that are used throughout the OGC Sensor Web Enablement (SWE) framework.

Plug-N-Play, auto-configuring, and autonomous sensor networks - SensorML enables the development of plug-n-play sensors, simulations, and processes, which seamlessly be added to Decision Support systems. The self-describing characteristic of SensorML-enabled sensors and processes also supports the development of auto-configuring sensor networks, as well as the development of autonomous sensor networks in which sensors can publish alerts and tasks to which other sensors can subscribe and react.

Archiving of Sensor Parameters - Finally, SensorML provides a mechanism for archiving fundamental parameters and assumptions regarding sensors and processes, so that observations from these systems can still be reprocessed and improved long after the origin mission has ended. This is proving to be critical for long-range applications such as global change monitoring and modeling.

Within SensorML, everything including detectors, actuators, filters, and operators are modeled as processes. The type of those processes is either physical or non-physical. The former are called *ProcessModels*, the latter *Components*. Physical processes define hardware assets where information regarding location or interface matters, non-physical processes define merely mathematical operations. The composite pattern allows the composition of complex physical and non-physical processes, called *ProcessChains* and *Systems*. All process types are derived from an AbstractProcess that defines the *inputs*, *outputs*, and *parameters* of that process, as well as a collection of metadata useful for discovery and human assistance. The inputs, outputs, and parameters are all defined using SWE Common data types. Process metadata includes identifiers, classifiers, constraints (time, legal, and security), capabilities, characteristics, contacts, and references, in addition to inputs, outputs, parameters, and system location. Further on, it allows modeling the life span of a specific process by defining its history in the form of an event list (e.g. recalibration, adjustments, etc.events).

 Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

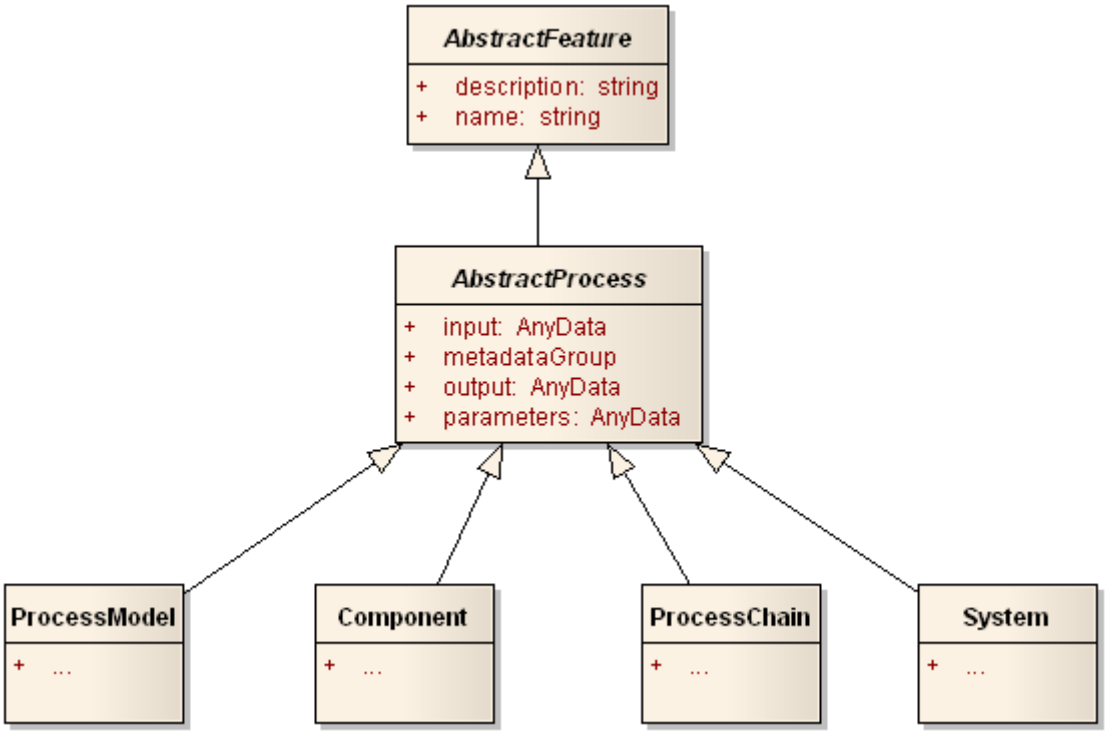


Figure 2. SensorML process types

The individual processes, as well as data sources (processes with no inputs), can be linked within a *ProcessChain* such that one can describe either the process by which an observation was derived (i.e. its lineage) or a process by which additional information can be derived from an existing observation. The general idea behind this concept is that one can re-use ProcessChains defined externally as part of the own process chain. Thus, complex chains only have to be defined once and can be re-used when made available online. The definition of links allows the proper lineage of a process chain, i.e. describes the relationships between individual processes of the chain.

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

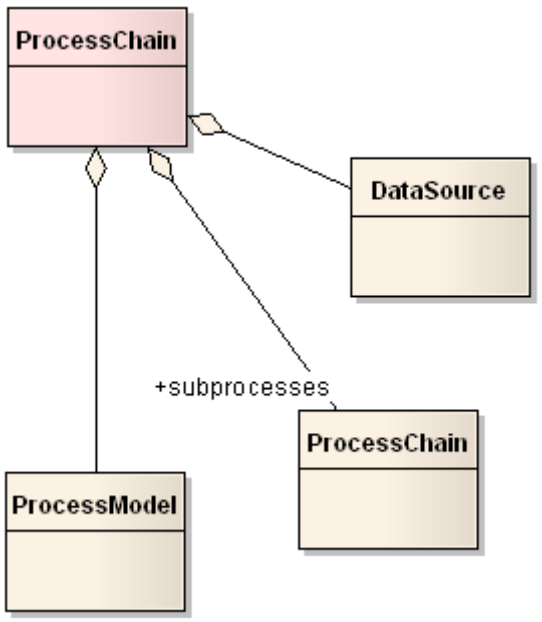


Figure 7-1. ProcessChain aggregates process models, other process chains, or data sources

System is a physical equivalent of a *ProcessChain.* It allows one to relate one or more processes to the "real world" by allowing one to specify relative locations and data interfaces. A System may include several physical and non-physical processes. In addition to the individual process of a process chain and its relationship to each other in terms of out-input behavior, a System defines the position of each component, i.e. it allows describing one physical asset in relation to another one.

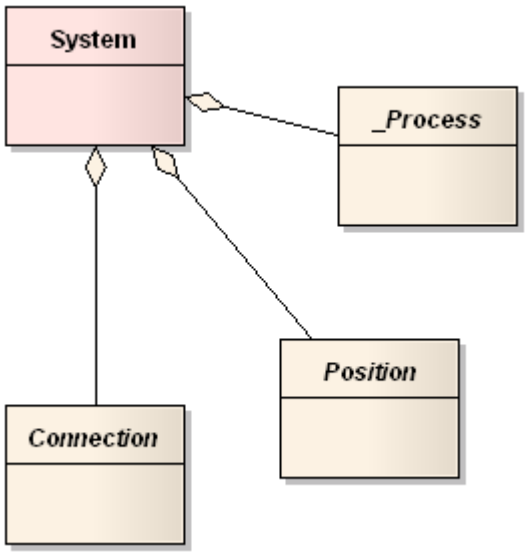


Figure 7-2. SensorML System aggregates processes, components, or data sources and describes their position and connnection

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

A public forum for SensorML is actively available at
http://mail.opengeospatial.org/mailman/listinfo/sensorml.

7.3 Observations and Measurements (O&M) [OGC 07-022 & OGC 07-002r2]

The general application of a sensor is to observe one or many properties in its environment. Each property belongs to a specific feature that represents an identifiable object. Applying the more general sensor model as described in clause 5.2, sensors observe properties of features and produce values for those properties. The observation itself is modeled as feature as well.

To reflect the general difference between the observation itself, i.e. the act of producing a value for a property of a feature, and the sampled feature itself, Observations and Measurements consists of two parts: Part one, *Observation schema* (OGC 07-022), describes a conceptual model and encoding for observations and measurements. This is formalized as an Application Schema, but is applicable across a wide variety of application domains. Part two, *Sampling Features* (OGC07-002r2), describes a conceptual model and encoding for the feature that has been observed. According to O&M, every observed property belongs to a feature of interest. Though often treated as identical and mostly of little interest to the consumer of the observation data, there is a conceptual difference between the *Sampled Feature* and the *Feature of Interest* of an observation. The difference is described best using some examples:

- In remote sensing campaigns, the sampled feature is a scene or a swath, whereas the feature of interest often defined as a parcel, a region, or any other form of geographically bounded area

- In-situ observation campaigns may obtain the geology of a region at outcrops (sampled features), whereas the feature of interest is the region itself

- Meteorological parameters might get sampled at a station, whereas the feature of interest is - strictly spoken - the world in the vicinity of that station

As said before, those differences are often of interest to members of the sampling campaign only. From a data consumer's point of view, who is not analyzing the data for potential errors in the sampling itself, sampling feature and feature may be identical.

The term *Measurements* in *Observation & Measurements* reflects the fact that most sensors produce estimates for physical quantities, i.e. for measures. Thus, a measurement is a specialized observation. This is somewhat is contrast to the conventional measurement theory, but inline with discussions in recent publications.

O&M 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. The observation is modeled as a Feature within the context of the General Feature Model [ISO 19101, ISO 19109]. An observation feature binds a result to a feature of interest, upon which the observation was made. The observed property is a property of the feature of interest, as illustrated in the following figure.

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

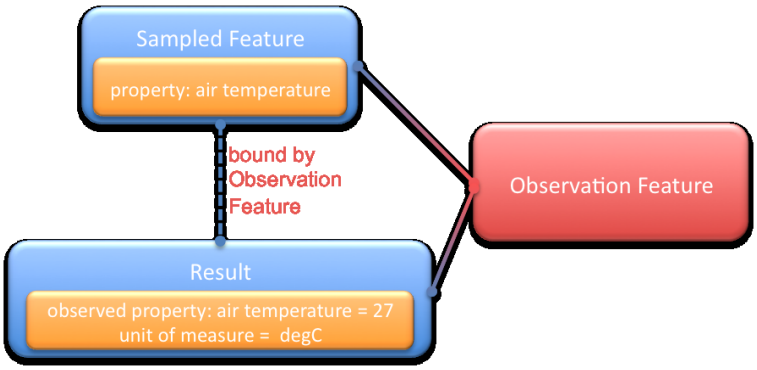


Figure 7-3: Binding of observation results to properties of the sampled feature

An observation uses a procedure to determine the value of the result, which may involve a sensor or observer, analytical procedure, simulation or other numerical process. This procedure would typically be described as a process within SensorML. The observation pattern and feature is primarily useful for capturing metadata associated with the estimation of feature properties, which is important particularly when error in this estimate is of interest.

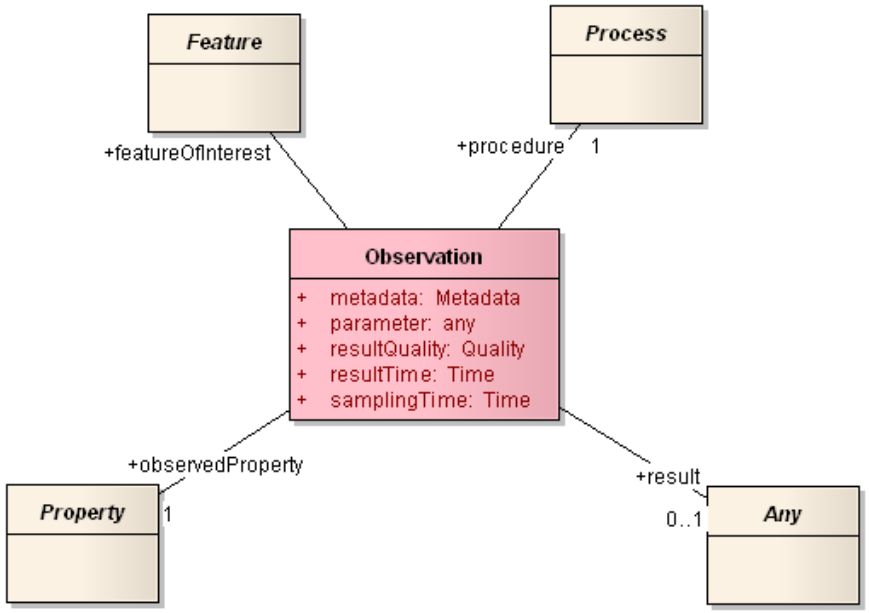


Figure 4. Simplified structure of the O&M *Observation* describing the featureOfInterest, the applied process, the observedProperty (which is a characteristic of the featureOfInterest), the result and the quality of the result value, the samplingTime (time the sample was taken) and resultTime (time the result was produced, which might be the same as samplingTime), additional metadata, and parameters describing the observation event that are not tightly bound to the process or featureOfInterest.

Generally, the result of an observation is of type “any”, i.e. it may be modeled in any form using any encoding. This approach allows a maximum of flexibility. Information

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

communities may define their own result models and use them within their community. From a universal point of view, this flexibility may prevent non-members of those communities to make any sense out of the result data. It is still an area of current research to define styles that adequately and efficiently support both simple and complex results, as well as perhaps legacy formats and out-of-band data. These observation models typical differ only in the encoding of the *resultDefinition* and *result* properties. All other properties of the *Observation* model remain common for all derived observation types. Therefore the O&M specification allows for extension of the *Observation* object to support various styles of providing observation result values and defines a number of Observation specializations that cover the most frequently used observation-types.

Experiences have shown that users usually use those specializations and may amend them with community-specific extensions, rather than defining result models directly derived from *Observation*. This strategy simplifies the identification of certain key elements of the result set and allows some processing of the result data, e.g. for visualization purposes, without understanding the full result model. Parsers simply skip the extensions.

O&M defines the following specializations of the generic observation (in a separate namespace named "omx" to avoid clashing of XML Schema imports):

For single-value properties, the observation feature is modeled as

- *CountObservation*, if the result is an integer representing the count of the observed property
- *CategoryObservation*, if the result is a textual value from a controlled vocabulary
- *TruthObservation,* if the result is a boolean value representing the truth value (usually existence) of the observed property
- *GeometryObservation*, if the result is a geometry
- *TemporalObservation*, if the result is a temporal object
- *ComplexObservation*, if the result is a record representing a multi-component phenomenon
- *Measurement*, if the result is a Measure, i.e. the result is a value described using a numeric amount with a scale or using a scalar reference system

If the values of the property vary over time and/or space, then the observation-type is modeled as discrete coverages of type:

- *PointCoverageObservation*, if the result is a point coverage which samples properties at points in the feature of interest,
- *DiscreteCoverageObservation*, if the result is a generalized discrete coverage

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

- *TimeSeriesObservation*, if the result is a time-instant coverage which samples a property of the feature of interest at different times

- *ElementCoverageObservation*, if the result is a coverage whose domain elements contain references to objects encoded elsewhere, which provide the sampling geometry of the feature of interest

Through the O&M specification, the SWE framework provides a standard XML-based package for returning observation results. Using a standard package in which to download observations from an SOS alleviates the need to support a wide range of sensor-specific and community-specific data formats. To achieve an even higher level of interoperability, SWE Common shall be used to fill in the result slot of an observation.

7.4 Transducer Model Language (TML) [OGC 06-010r6]

Transducer Markup Language (TML) is a method and message format for describing information about transducers and transducer systems and capturing, exchanging, and archiving live, historical and future data received and produced by them. A transducer is a superset of sensors and actuators. TML provides a mechanism to efficiently and effectively capture, transport and archive transducer data, in a common form, regardless of the original source. Having a common data language for transducers enables a TML process and control system to exchange command (control data) and status (sensor data) information with a transducer system incorporating TML technology. TML utilizes XML for the capture and exchange of data.

TML was designed with the express goal of facilitating the development of a "Common" Transducer Processing/Control machine while also facilitating interoperable machine-to-machine communications. For the purposes of data fusion and post analysis, it is paramount to preserve raw transducer data in as close a manner to the original form as possible. Although data would be ideally preserved in its raw format, it is impossible in some cases to do so. TML provides facilities to capture data at any stage, from raw production, to partially processed, to final data forms. Greater benefits of TML are realized the closer to the source raw data one gets.

Transducer Markup Language (TML) defines:

- a set of models describing the hardware response characteristics of a transducer.

- an efficient method for transporting sensor data and preparing it for fusion through spatial and temporal associations

Sensor data is often an artifact of the sensor's internal processing rather than a true record of phenomena state. The effects of this processing on sensed phenomena are hardware-based and can be characterized as functions.

TML response models are formalized XML descriptions of these known hardware behaviors. The models can be used to reverse distorting effects and return artifact values to the phenomena realm. TML provides models for a transducer's latency and integration times,

 Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

noise figure, spatial and temporal geometries, frequency response, steady-state response and impulse response.

Traditional XML wraps each data element in a semantically meaningful tag. The rich semantic capability of XML is in general better suited to data exchange rather than live delivery where variable bandwidth is a factor. TML addresses the live scenario. The TML cluster is a terse XML envelope designed for efficient transport of live multiplex sensor data. It also provides a mechanism for temporal correlation to other transducer data.

In November 2005, the TML specification document OGC 05-085 was approved as a Public Discussion Paper. In February 2006, TML document OGC 06-010 was submitted as a Pending Document to the Huntsville Technical Committee (March 2006) to begin the RFC process as an OGC Technical Specification. Eventually, version OGC 06-010r6 was released as OGC Standard. Further information on TML can be found at http://www.transducerml.org.

7.5 Sensor Observation Service (SOS) [OGC 06-009]

The goal of SOS is to provide access to observations from sensors and sensor systems in a standard way that is consistent for all sensor systems including remote, in-situ, fixed and mobile sensors. This is a challenging task because the users of sensor data have historically been divided into those who primarily deal with in-situ sensors and those who primarily deal with remote sensors. The terminology, perspective, and expectations of these two broad groups are different. SOS leverages the Observation and Measurements (O&M) specification for modeling sensor observations and the SensorML specification for modeling sensors and sensor systems.

An SOS organizes collections of related sensor system observations into Observation Offerings. The concept of an Observation Offering is equivalent to that of a sensor constellation discussed earlier in this document. An Observation Offering is also analogous to a “layer” in Web Map Service because each offering is intended to be a non-overlapping group of related observations. Each Observation Offering is constrained by a number of parameters including the following:

- Specific sensor systems that report the observations,
- Time period(s) for which observations may be requested (supports historical data),
- Phenomena that are being sensed,
- Geographical region that contains the sensors, and
- Geographical region that is the subject of the sensor observations (may differ from the sensor region for remote sensors)

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

The approach that has been taken in the development of SOS, and the SWE specifications on which it depends, is to carefully model sensors, sensor systems, and observations in such a way that the model covers all varieties of sensors and supports the requirements of all users of sensor data. This is in contrast to the approach that was taken with the Web Feature Service (WFS). WFS provides a generic definition of a geographic feature that is flexible enough to encompass any real-world entity. The WFS uses GML application schemas to define the specific properties of each type of feature. With this approach, interoperability requires organizations to agree on domain-specific GML application schemas. Clients that access a WFS in a particular domain must have a-priori knowledge of the application schemas used in that domain. The SOS approach defines a common model for all sensors, sensor systems and their observations. This model is not domain-specific and can be used without a-priori knowledge of domain-specific application schemas.

In November 2005, the SOS specification document OGC 05-088r1 was approved as a Public Discussion Paper. In February 2006, SOS document OGC 06-009 was submitted as a Pending Document to the Huntsville Technical Committee (March 2006) to begin the RFC process. Eventually, OGC document 06-009r6 was approved as an OGC Standard.

7.6 Sensor Alert Service (SAS) [OGC 06-028r5]

The Sensor Alert Service (SAS) can be compared with an event stream processor in combination with an event notification system. An SAS processes incoming sensor data continuously. Pattern-matching algorithms identify satisfied alert conditions and start the alert distribution process.

An SAS might therefore provide a wide variety of alerts related to sensors and sensor observations including, as examples, measured values above a threshold, detected motion or the presence of a recognizable feature, or perhaps sensor status (e.g. low battery, shutdown or startup). SAS is a transactional service in two counts: First sensors can dynamically connect to the service and publish metadata and observation data (using SensorML and O&M). Sensors don't have any knowledge about current subscriptions, as SAS doesn't provide such information. Second, event consumers may define their own event conditions. For every new event type definition, the SAS is opening a new distribution channel and incorporates the alert type in its capabilities documentation, i.e. every new alert type, defined by an alert consumer, is available to other users as well.

An SAS can *advertise* what alerts it can provide. A consumer (interested party) may *subscribe* to alerts disseminated by the SAS. If an event occurs the SAS will *publish* an alert and *notify* all clients subscribed to this event type through a messaging service. Currently, SAS supports XMPP (Extensible Messaging and Presence Protocol) for alert distribution exclusively, although in combination with a Web Notification Service (WNS), it may deliver alerts to other communication endpoints as well. This pattern is likely to change in future versions of the SAS. Currently, the SAS editors are busy doing research on alternative alert and notification mechanisms and protocols.

 Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

The SAS specification has been developed under an Interoperability Experiment. Currently, OGC document 06-028r5 is published as an OGC Best Practice Paper. The RFC process was finished end of 2007. The next steps are the formation of a Standard Working Group to get SAS released as on OGC standard by the end of 2008, beginning of 2009.

7.7 Sensor Planning Service (SPS) [OGC 07-014r3]

The Sensor Planning Service (SPS) is intended to provide a standard interface to collection assets (i.e., sensors, and other information gathering assets) and to the support systems that surround them. Not only must different kinds of assets with differing capabilities be supported, but also different kinds of request processing systems, which may or may not provide access to the different stages of planning, scheduling, tasking, collection, processing, archiving, and distribution of requests and the resulting observation data and information that is the result of the requests. The SPS is designed to be flexible enough to handle such a wide variety of configurations.

SPS uses SWECommon to describe planning parameters that have to be set by users. SPS is often used together with WNS and SOS. The interaction sequence basically consists of the following steps:

- Discovery of SPS as a service that is capable of providing required data sets (given that the required data was not provided by an SOS already).
- Requesting the list of parameters that have to be set in order to start the SPS process (the backend of an SPS is opaque to the user, but described using SensorML, i.e. the user is provided with all information about the process itself, but gets no information about the communication between SPS and the process)
- Providing values for all required parameters and execution of the process
- Receiving information about the availability of the produced data at an OGC interface, e.g. SOS

The SPS specification document [OGC 07-014r3] was approved as an OGC Standard in 2007. Currently, an SPS Standard Working Group is formed to improve the SPS standard.

7.8 Web Notification Service (WNS) [OGC 06-095r1]

Web Service environments provide a suitable method to gather requested information in an appropriate way. Synchronous transport protocols such as HTTP provide the necessary functionalities to post requests and to receive the respective responses. HTTP is a reliable protocol in the way it ensures the packet delivery, in order, and with a definitive acknowledgement for each delivery or failure. In case of a simple Web Map Service, a user will receive visualized geographic information after a negligible amount of time, or the

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

user will receive an exception message. HTTP satisfies the needs for this kind of processing, without the need for further functionality.

As services become more complex, basic request-response mechanisms need to contend with delays/failures. For example, mid-term or long-term (trans-) actions demand functions to support asynchronous communications between a user and the corresponding service, or between two services, respectively. A Web Notification Service (WNS) is required to fulfill these needs within the SWE framework.

The Web Notification Service Model includes two different kinds of notifications. First, the "one-way-communication" provides the user with information without expecting a response. Second, the "two-way-communication" provides the user with information and expects some kind of asynchronous response. This differentiation implies the differences between simple and sophisticated WNS. A simple WNS provides the capability to notify a user and/or service that a specific event occurred. In addition, the latter is able to receive a response from the user.

The basis on which notifications will be sent is free to the service and will be described in its capabilities. The "way-of-notification" palette may include:

- e-mail
- http-call (as HTTP POST: in case of sophisticated clients that act as web services themselves)
- SMS
- Instant Message
- phone call
- letter
- fax
- XMPP
- any other communication protocol

Once a client registers a user along with the method of notification desired, the client receives a unique RegistrationID that can then be provided as input to other services (e.g. SPS or SAS).

The WNS specification document [OGC 06-095r1] was approved as a OGC Best Practice Paper in 2007. A Standard Working Group will be formed next to start the development of of the OGC Standard. The release of WNS as an OGC standard is likely to appear early 2009.

7.9 Sensor Web Registry

The Sensor Web registry is still in an early testing phase. It was implemented successfully using an OGC Catalog Service backed up by an ebRIM/ebXML engine during OWS4 in 2006. This service provides discovery capability throughout the whole sensor web infrastructure. Typical requests to this service are 'GetRecords' operations containing fil-

 Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

tering parameters used to search a database for one or more matching objects of interest. These objects include SWE services (as well as other OGC services), sensor descriptions, process chains and dictionary entries such as phenomena or units, etc.

In order to be able to insert objects to a catalogue, each object type must be defined by a schema and a CSW harvest profile. This profile shall define what information needs to be parsed out of the object XML and advertised as searchable content.

The following table shows what pieces can be mined from different XML documents used through the SWE framework:

Document Name	Searchable Sections/Tags
SOS Capabilities	OWS common section (like any other service) For each observation in the offering list: - observation id, name and description - observed property (association with O&M phenomenon object) - procedure id (association with SensorML sensor object) - feature of interest (association with GML feature) - time range - location (if fixed) - format
SPS Capabilities	OWS common section (like any other service) For each sensor system in the offering list: - phenomenon urn (association with O&M phenomenon object) - sensor id (association with SensorML sensor object) - area of service
SAS Capabilities	OWS common section (like any other service) For each subscription in the offering list: - alert id, name and description - observed property (association with O&M phenomenon object) - procedure id (association with SensorML sensor object) - feature of interest (association with GML feature) - time range - location (if fixed) - format
SensorML Sensor, System and Process	Most information is contained in the metadata group - description - identifiers - classifiers - time, legal and security constraints - characteristics - capabilities - contacts - inputs and outputs (association with O&M phenomenon) - taskable parameters (association with O&M phenomenon) ➔ eventually recurse for each sub components
SWE Phenomena	A phenomenon is intended to be a pure dictionary entry, so it should be parsed in its entirety, including:

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

	- description - name - type - constraint value - component if composite (association with other SWE phenomenon)

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

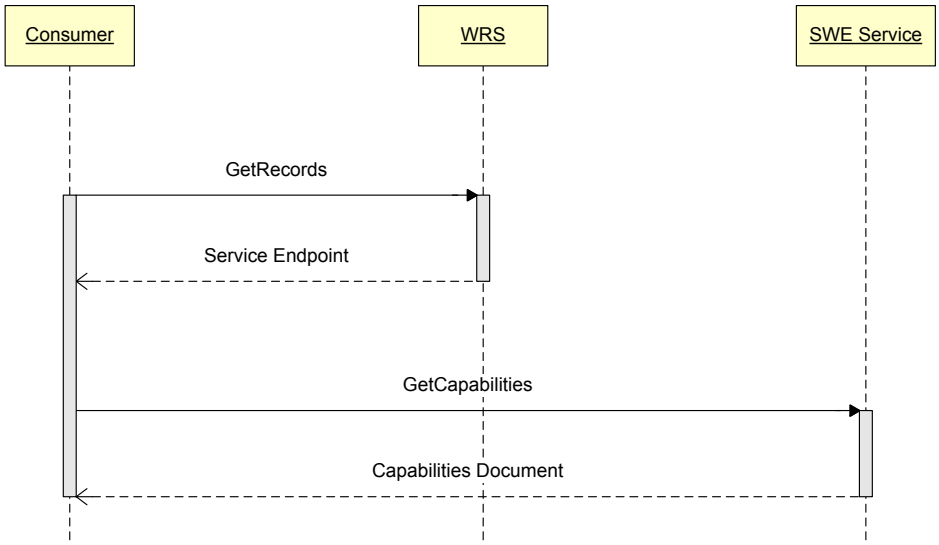
Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

8 Typical Work Flows

The following diagrams illustrate simple workflows involving one or several SWE services. This section covers service discovery and registration with a registry, access to discrete and streaming sensor data, sensor tasking, publication of alerts and subscription to alert services. More advanced diagrams show a specific SWE configuration using simple sensor nodes connected via a private network to a data center where data can be archived, processed and made available on the public Internet.

8.1 Discovery of SWE Services using a Registry

This first scenario shown below demonstrates how SWE services, like any other OGC web service can be discovered using a registry. In this case, the consumer (which can be a client or another service) first connects to a WRS and sends a search query for a specific type of SWE service using a 'GetRecords' message. The WRS looks for a matching record in its database and returns an XML document containing the endpoint URL of all matching services found. The client is then able to connect to the service of interest and request the complete capabilities document.

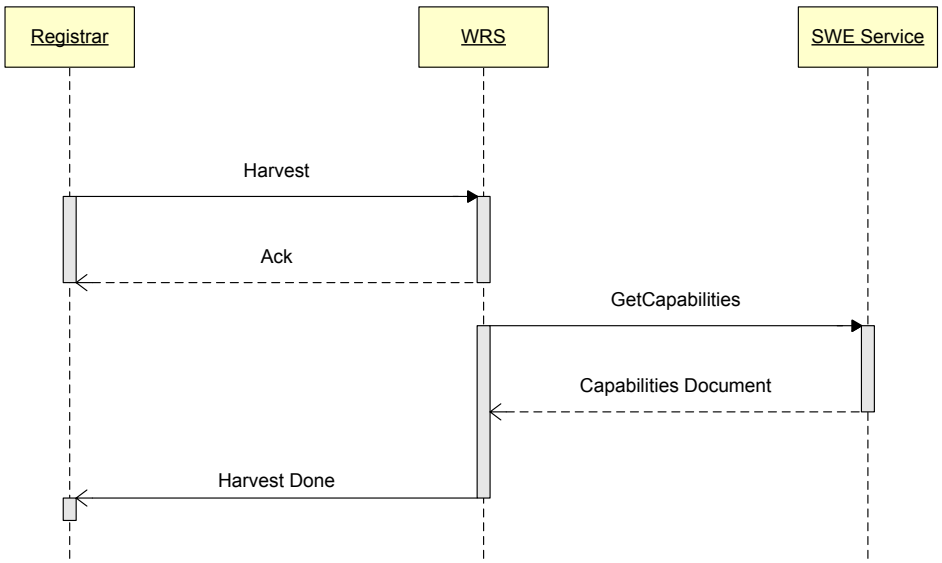


SWE Service discovery sequence

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

8.2 Registration of a new SWE Service in a Registry

The following scenario shows how to register a new service to a registry. The registrar, which can be the service itself or a third party, sends a 'Harvest' command to the WRS containing the endpoint URL of the service to be registered. The 'Harvest' operation is then executed asynchronously by the WRS when resources are available. When so, the WRS connects to the specified service, fetches its capabilities document, and starts processing it according to the CSW profile defined for this type of registry object. This profile defines what part of the document should be parsed and used as searchable fields in the registry. When the 'Harvest' operation is complete, the WRS sends a notification message to the registrar. The newly added service is now searchable through the WRS interface.



SWE Service registration sequence

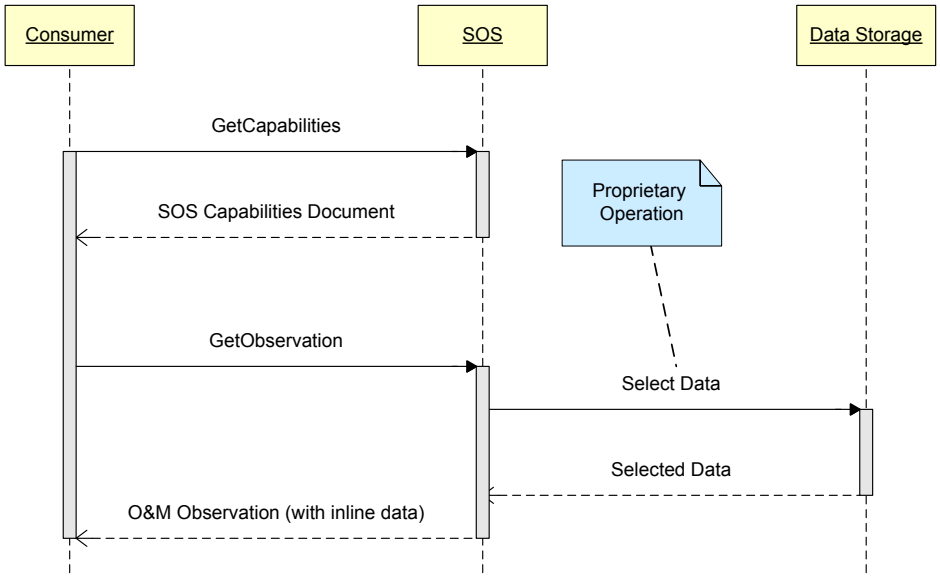
(Note that the Registrar can be on the same node as the Service itself)

It is important to realize that each service type should provide a CSW profile in order to be successfully registered by a WRS. A default profile can nevertheless be created using only the common OWS section used by all new OGC services. This will provide a minimum set of searchable information, put doesn't allow to discovery of individual layers or offerings which are service specific.

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

8.3 Request of discrete observation data

This sequence diagram shows the simple process of requesting observation data from an SOS service. The consumer (perhaps after having discovered the service) issues a 'GetCapabilities' request to obtain a list of all observation offerings available on this server. Using this information, the consumer then selects a specific offering, one or more phenomena, a finite time range and perhaps a region of interest (bbox) and sends a 'GetObservation' command with these parameters to the SOS. The SOS then internally reads the requested data using proprietary technology (perhaps a set of files or a DB) and sends it back to the consumer encapsulated in an O&M observation, which provides metadata as well as temporal and spatial characteristics of the data.



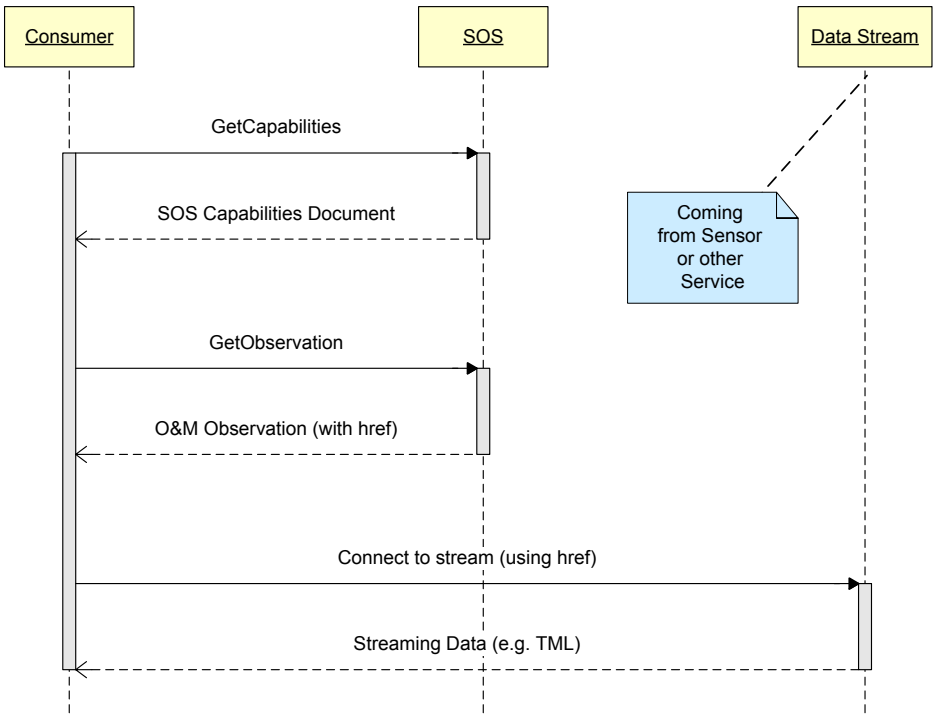
SOS GetObservation for archived data

In this example, the request is based on a finite time range selecting archive data. The size of the block of data is therefore known and it is possible to include it inline within the O&M document. The specially designed SWE response formats allows the structure and encoding of the data to be fully described, thus allowing base64 encoded data within the XML in order to support heavy datasets such as imagery or radar data (Note that raw binary is also available if providing out of band data).

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

8.4 Request of streaming (out of band) observation data

This diagram demonstrates how streaming data (eventually real time) can also be accessed through an SOS interface. The consumer first downloads the capabilities document which may contain an offering tagged as real time data. If this is the case, the consumer can issue a 'GetObservation' request for this offering and a time range specifying an end date in the future. The consumer can then start receiving real time data from the SOS or even directly from the source as indicated in the following diagram. In order to achieve this, the data is not provided inline in the O&M observation like in the previous example, but rather through a hyperlink (href) to the data source. This hyperlink can be a call to the SOS 'GetResult' operation which would return only the data, or another (proprietary) static URL or service request.



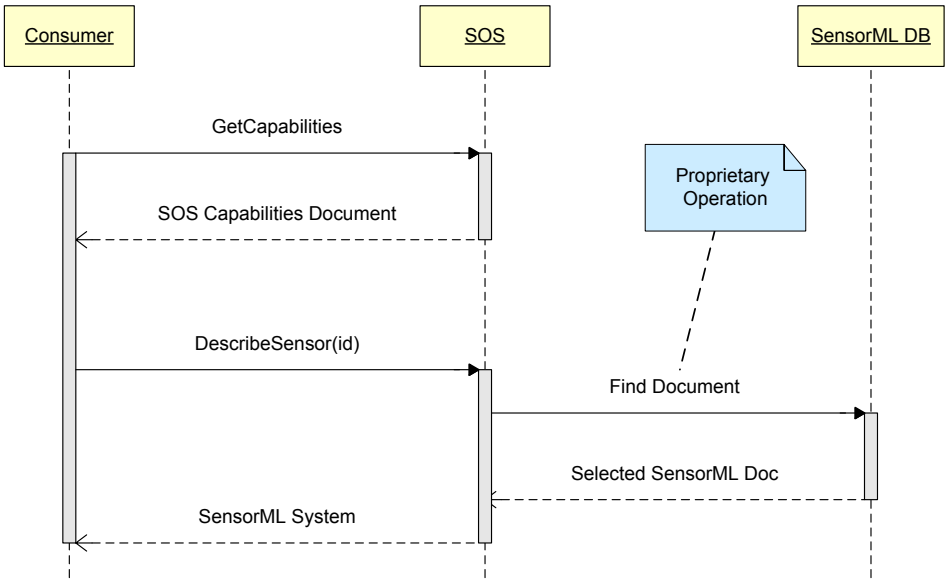
SOS GetObservation for real time streaming data

The O&M Observation still contains the description, structure and eventually encoding of the data, so that the consumer can process this information before starting to parse the incoming data stream. This means of accessing data can be used when pointing to any kind of out of band data, including raw binary and multiplexed streams (such as TML or AAF/ASF) when defined as such in the Observation.

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

8.5 Access to Sensor Descriptions

The SOS service also provides access to a SensorML description which defines the procedure by which observation were obtained. This procedure can be a sensor system or a simulator for instance. In order to accomplish this, the consumer of the following example first finds a sensor of interest advertised in the capabilities document provided by the service. It then issues a '*DescribeSensor'* request using the ID specified in the capabilities document. If the ID is valid, the SOS service shall then return the corresponding SensorML document after fetching it from a local database or any other proprietary system.



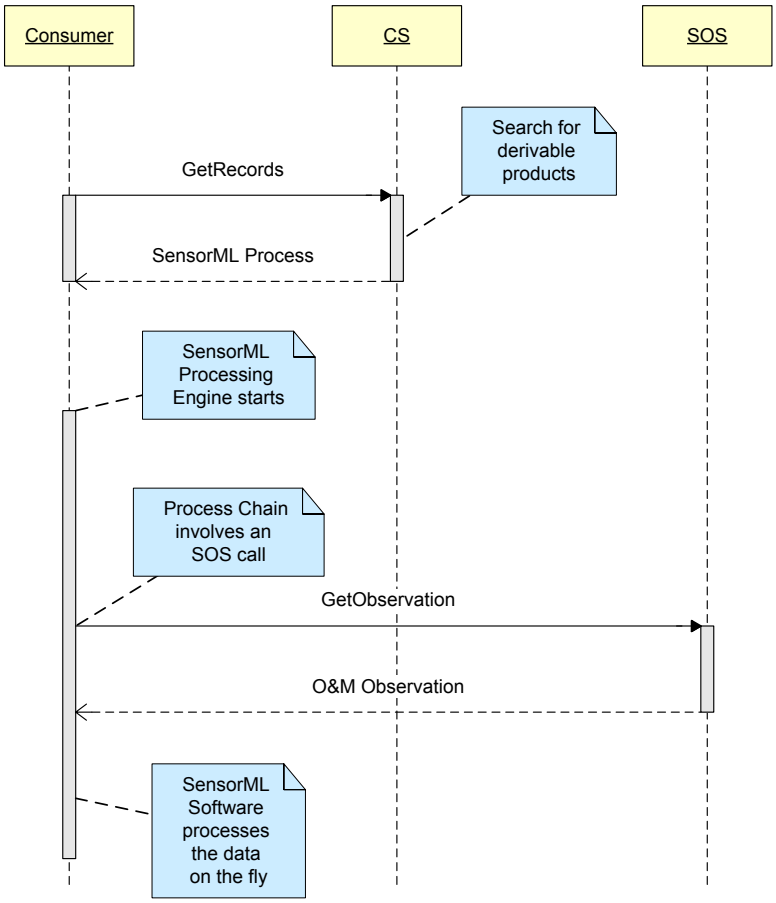
SOS DescribeSensor using SensorML documents DB

The SensorML document contains information about the sensor (or simulator) that was used to collect the observation data of the corresponding offering. This includes the calibration at the time of collection, but also the sensor location and eventually attitude. This information represents the lineage of observation data (i.e. what happened to the real phenomenon during the sampling and detecting stages of the measurement process) and is directly usable for processing of the observation data. It is typically used for deriving meaningful values from the raw observation data as well as geo-locating the measurements.

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

8.6 Data Processing using SensorML

SensorML can be used very easily to process observation data coming from an SOS. A catalog service will provide the entry point to look for a higher level products, such as geolocated observations or products derived from measurements of several sensors. These derived product have traditionally been available on a server where they have been pre-computed. SensorML enables on-demand processing on the client side by providing a detailed process chain that can be applied to existing data in order to derive more useful information. The next sequence shows how a specific SensorML process can be discovered and executed by the client using generic SensorML software. The process itself can involve a call to one or more SOS in order to obtain the source data. Advantages of such a mechanism are that only exactly the desired data need to be requested and processed, and that the process can be tweaked to meet exact needs.

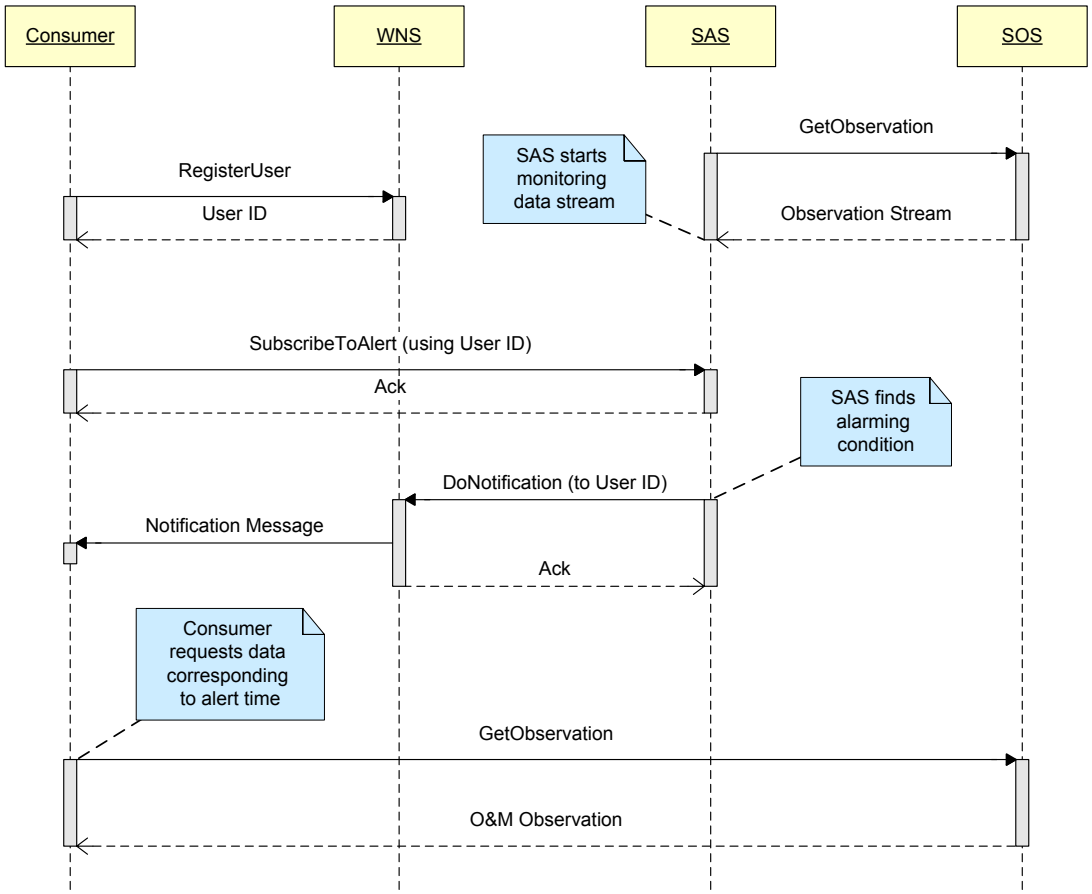


Processing SOS data using a SensorML Process Chain

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

8.7 Subscribing and Receiving Sensor Alerts

This diagram illustrates the use of three SWE services (SAS, SOS, and WNS) for subscribing to and receiving alerts generated by processing SOS data. The consumer first registers its contact information with a WNS, and gets back its user ID. The WNS is then responsible for sending future notification messages to the consumer using the provided means of communication. In the mean time a SAS service has been created and started monitoring a data stream for predefined alarming conditions. It is then assumed that the consumer somehow obtains the endpoint of this SAS service (for instance by issuing a search to a WRS). The consumer then connects to the SAS and subscribe to an alert advertised in its capabilities ('*GetCapabilities'* request not shown for clarity), using the desired WNS address and user ID. The SAS then acknowledges that the user is subscribed and should start receiving alerts when the given condition is met.



WNS - SAS - SOS collaboration for listening to alerts

When this certain condition is found in the data stream (e.g. value going over a threshold), the SAS contacts the WNS using a *DoNotification* command with the subscribed user ID and the message to be sent. The WNS then sends the notification message to the

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

consumer using the communication means (email, phone, service connection…) specified at the time of registration.

The last part of the diagram is optional and happens only if the notification message specifies the data source (here the SOS) from which the alert was derived. This allows the consumer to contact the SOS directly and request the data corresponding to the alert. This mode of operation is especially recommended when the data used to generate the alert is bulky and thus not suited to be included within the alert message itself (The alert message should be kept as small as possible since it has to be sent to all subscribers).

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

8.8 Tasking Sensor Systems

This more complex diagram shows the process by which sensor tasking is made possible using the SWE framework. This sequence assumes that the consumer has already registered its contact information with the WNS and obtained a user ID (see previous).

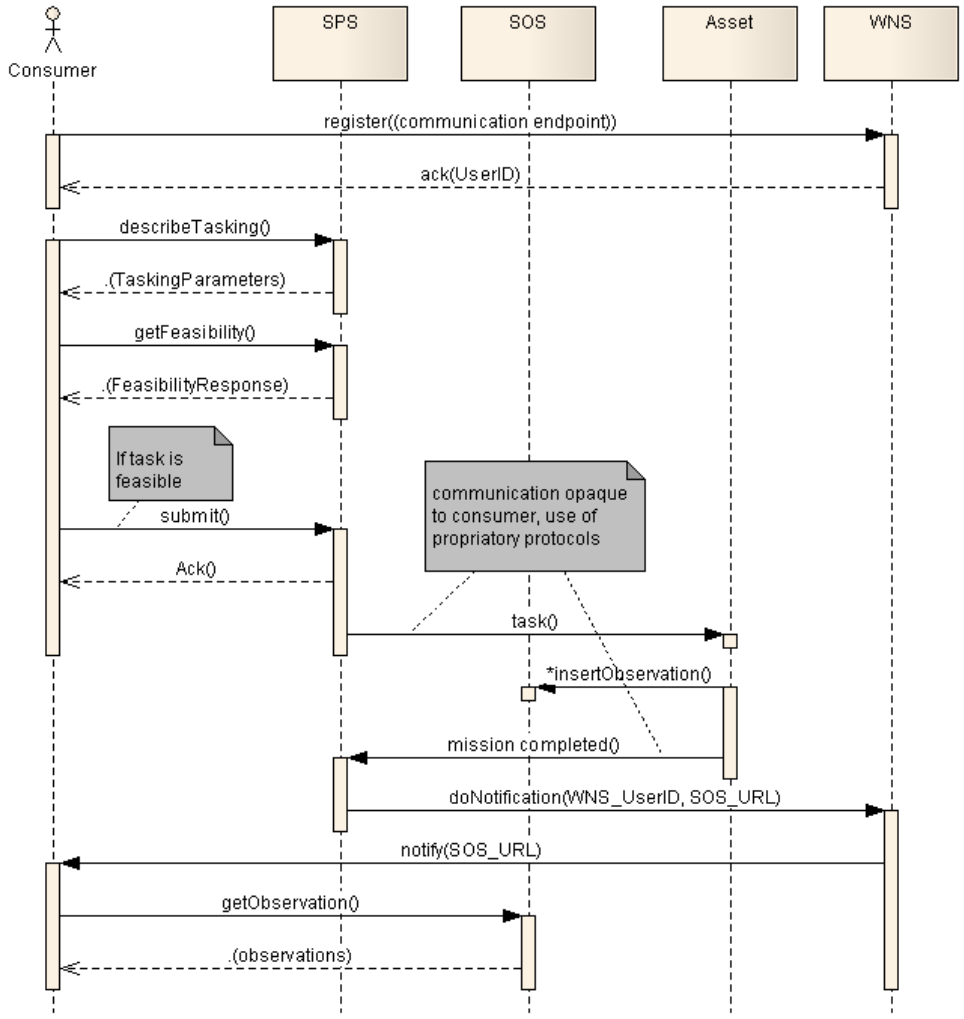


Figure 8-1: Typical sequence if data collection is requested

After registering with a WNS, the consumer issues a *DescribeTasking* request to the SPS and receives a detailed description of all taskable parameters. It then fills up desired values for these parameters and includes this in a *GetFeasibility* request in order to know if that particular task is doable with this service. If the SPS's feasibility response is positive (Note that the feasibility response can also be sent asynchronously using a WNS), the consumer can submit the task for execution. This *Submit* operation specifies the task parameters as well as the WNS/UserID to be used for notification of task completion. The

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

SPS internal software then tasks the corresponding asset to execute the mission, which is done asynchronously. While the measurement mission is executed, newly collected observations are inserted using the SOS (SOS-T) *InsertObservation* operation into an observation server (optionally, data might be put into a data storage directly). Upon completion of the task, the SPS (or the asset or a third party) issues a *DoNotification* command to the WNS and the consumer is in turn alerted of the availability of the requested data. The consumer then issues a *GetObservation* on the SOS that was used to collect the data.

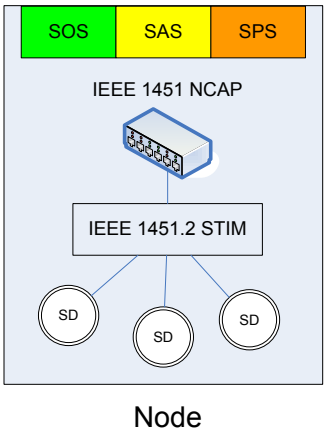
Real-Time low latency tasking is even possible by sending streaming commands to the SPS and receiving streaming data from the SOS, without using the notification mechanism.

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

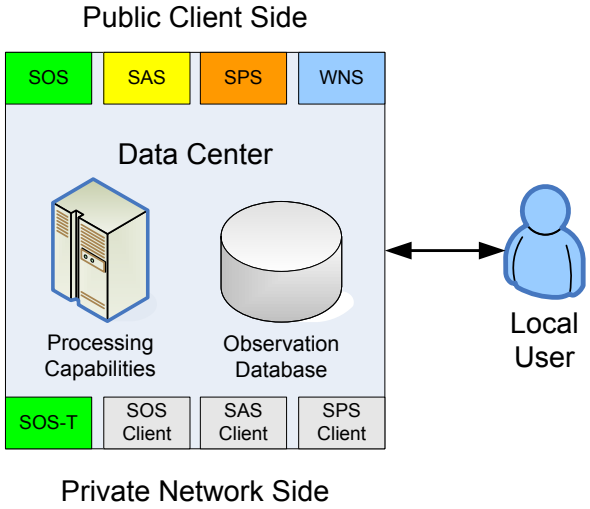
9 Implementation examples

9.1 Using SWE services in a "star" network configuration

The following diagrams show the use of SOS, SAS, SPS and WNS within a specific implementation of the SWE framework inspired by Oak Ridge National Labs SensorNet. This configuration involves multiple simple nodes connected through SWE services to a central data center with much greater processing power. Sensors connected to each node are IEEE-1451 sensors that have plug n' play capability. The following diagram illustrates the configuration of each node on the network.



Each node provides connectivity with other network components through optional SWE interfaces. Implementations of these interfaces are expected to be very light weight since the available processing power and storage capacity on each node is very low.

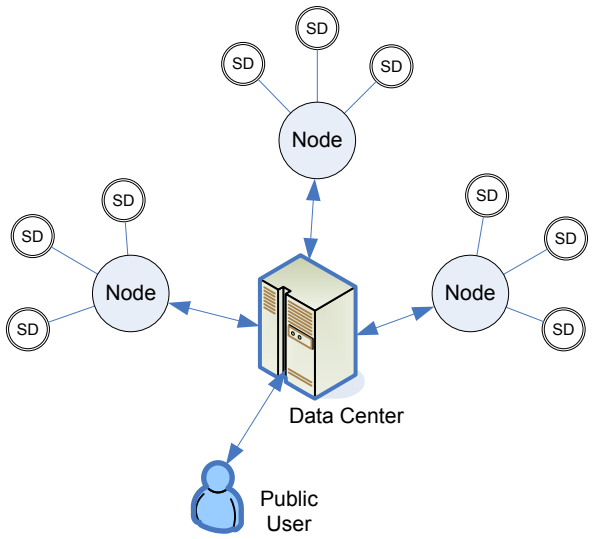


In this example, each node can either incorporate a small SOS providing access to real time streaming data (if network connection is permanent) or simply inserts its observation in the data center periodically (using if network connection is intermittent, e.g. GSM).

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

The SPS interface is only available on nodes that can achieve simple taskable operations like collecting data at a specific time or changing sensor parameters. The SAS is only available on nodes where a processing of the data is possible and used to automatically detect certain alarming conditions that will be reported to the data center. A simple WNS message generator (not the full service) is also available in order to send alert message or SPS notifications. The following diagram shows the configuration used for the data center, which is also used as a mission center (with SPS).

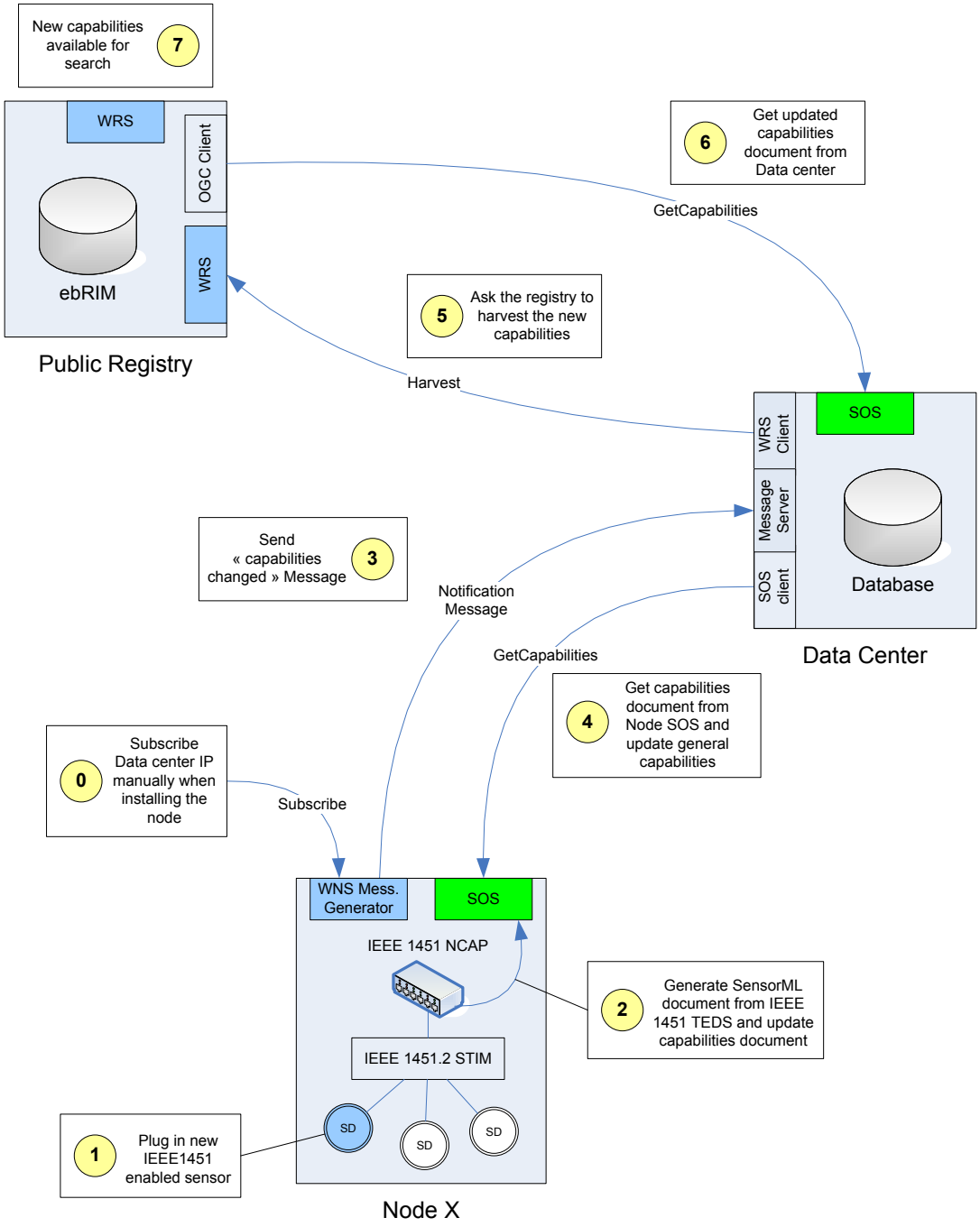
The data center provides many more features than a single node. It consists of a public interface accessible from the Internet and providing SOS, SAS, SPS and WNS services and a private interface only accessible from inside the sensor network and locally. The following diagram shows a global view of the network.



An unlimited number of nodes can of course be connected to the data center in this fashion, each of which reports data from multiple sensors. A public user doesn't have direct access to the nodes since the datacenter acts as a gateway. The intent with this kind of architecture is that the data center can reflect all individual nodes capabilities in one or more aggregate capabilities documents that are accessible to the public and this for all SWE services. The data center can also filter this information if some of it should not be made available to the public and eventually derive new products from raw sensor data that can be added to the global capabilities of the network.

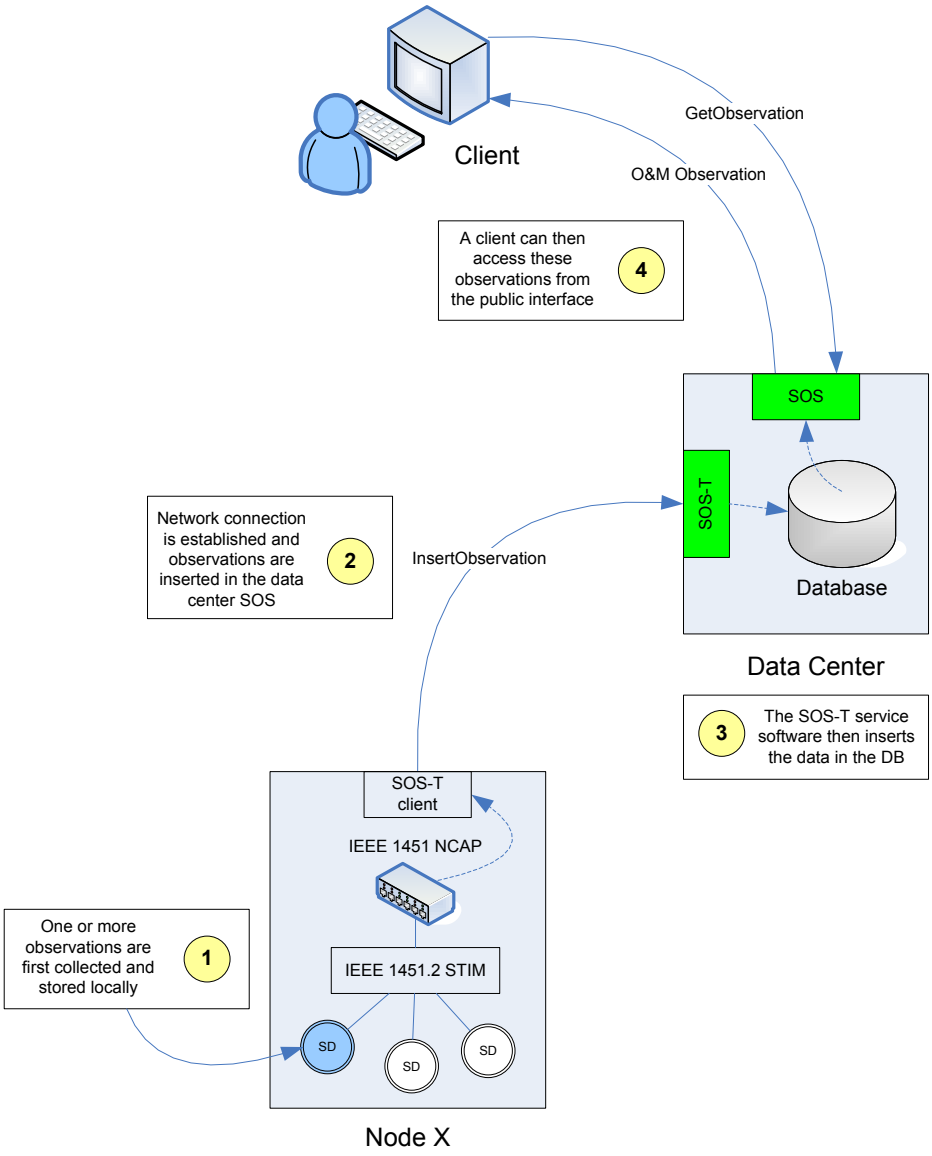
The following diagram shows how a new sensor is added to the system in a plug n' play fashion. This shows how capabilities documents and even the public registries are automatically updated after the new sensor is inserted in one of the nodes.

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.



The next diagram illustrates how individual nodes can insert there observations in the main database using the data center SOS-T interface. This way of operation is very useful when nodes cannot keep a permanent network connection with the data center (e.g. GSM).

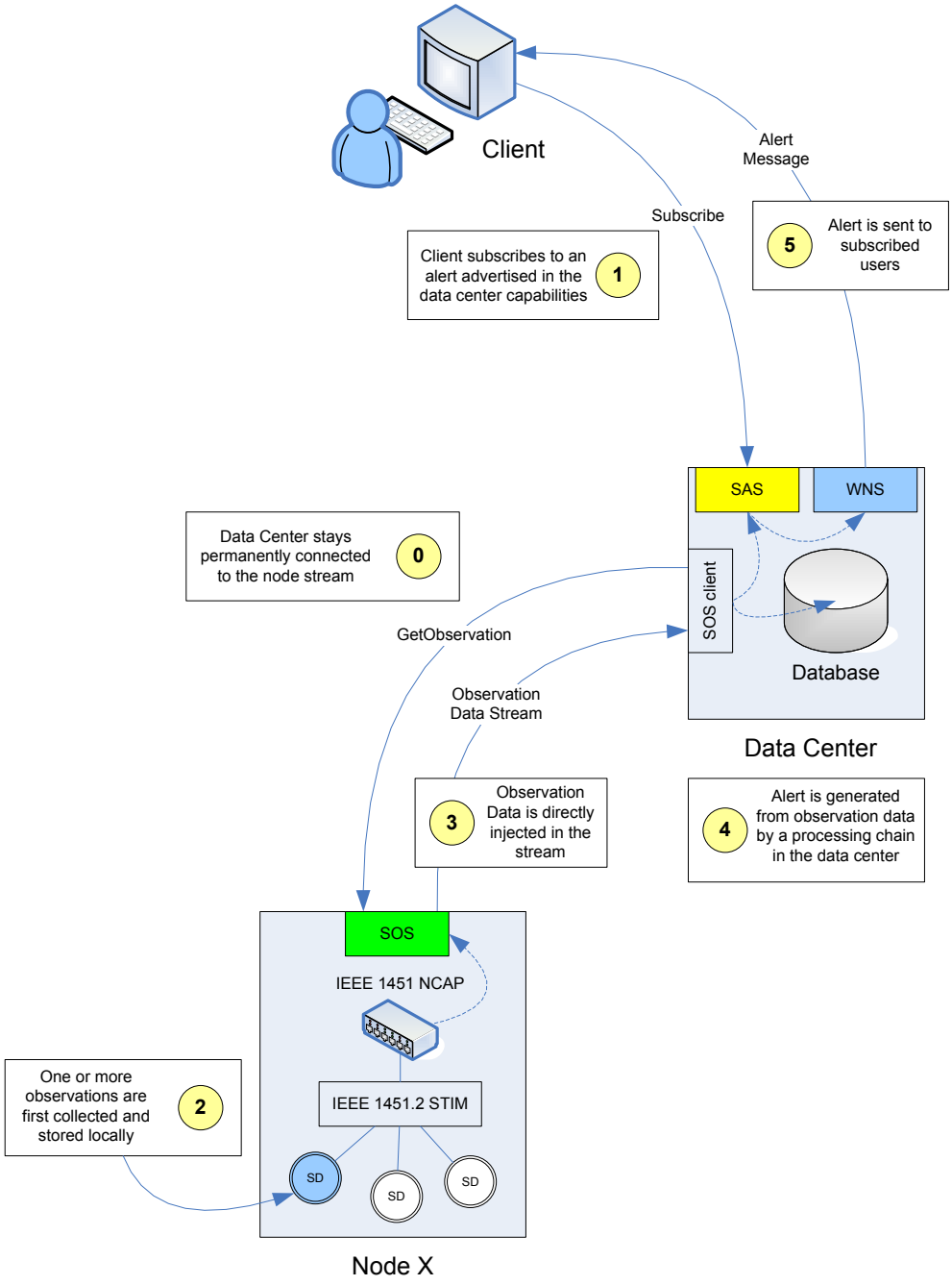
Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.



The new observations received at the data center can also feed a process chain used to generate advanced products, which can also be made available through the public interface. Some of these products can even be alerts generated by the data center and send to subscribers. This will be shown on the next diagram, but keep in mind that all these possibilities can be combined differently.

The following diagram shows the process by which nodes can stream their observation data directly to the data center when a permanent network connection is available. It also shows how an alert can be derived from these observations at the data center level and sent to all subscribed users.

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

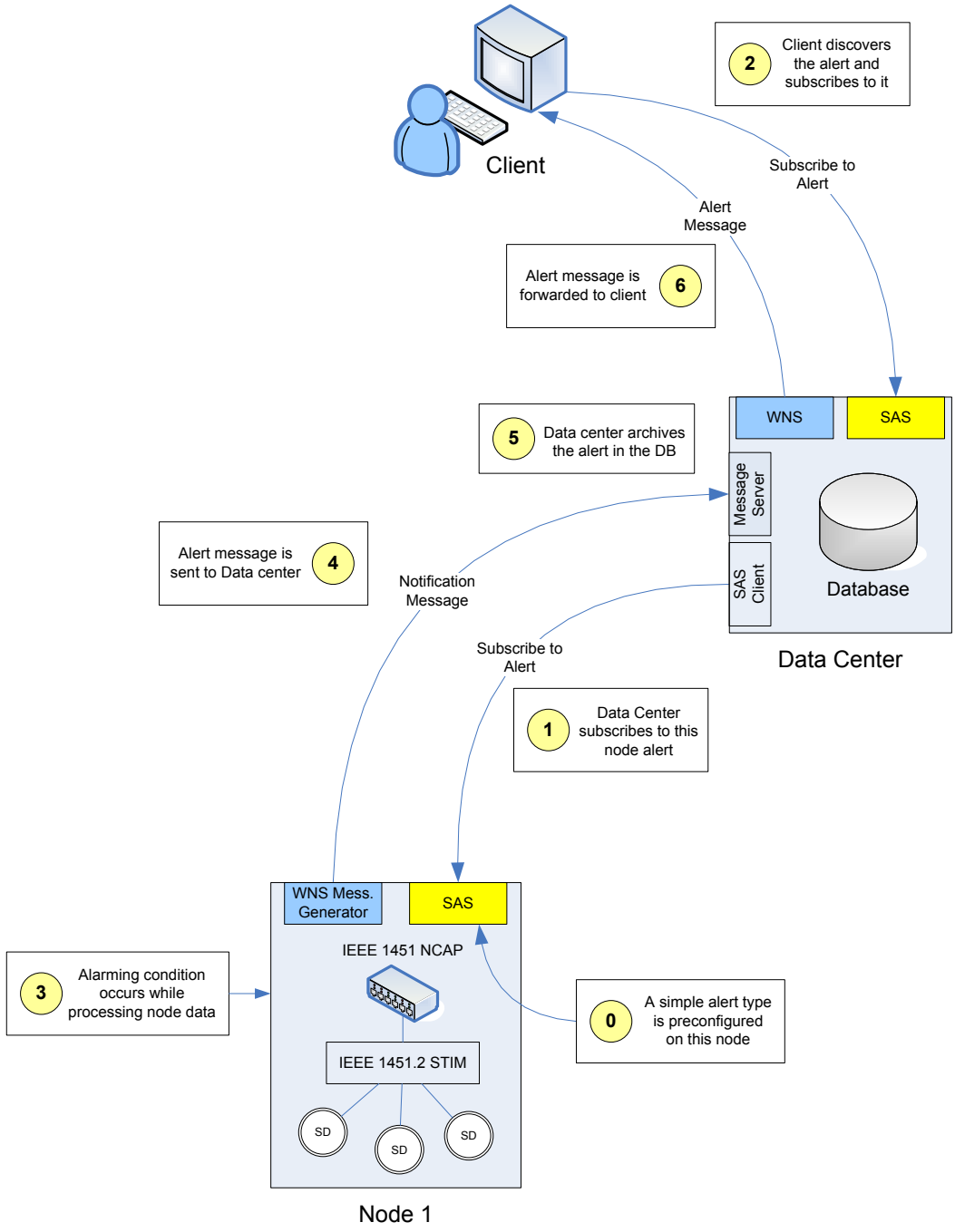


This streaming configuration allows data collected at the node to be uploaded to the data center even before the collection process is terminated. This is especially interesting if observations are needed in a close to real time manner or if the node capacity is too small to store the whole data (e.g. case of video). In parallel to the alert generation process, the data center can also archive the observation data in the main database just like it was done in the previous example. This is not shown on the diagram for clarity.

The next diagram shows how alerts can be generated at the node level and forwarded to the data center for archiving. Alerts can also be forwarded directly to the client if the later

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

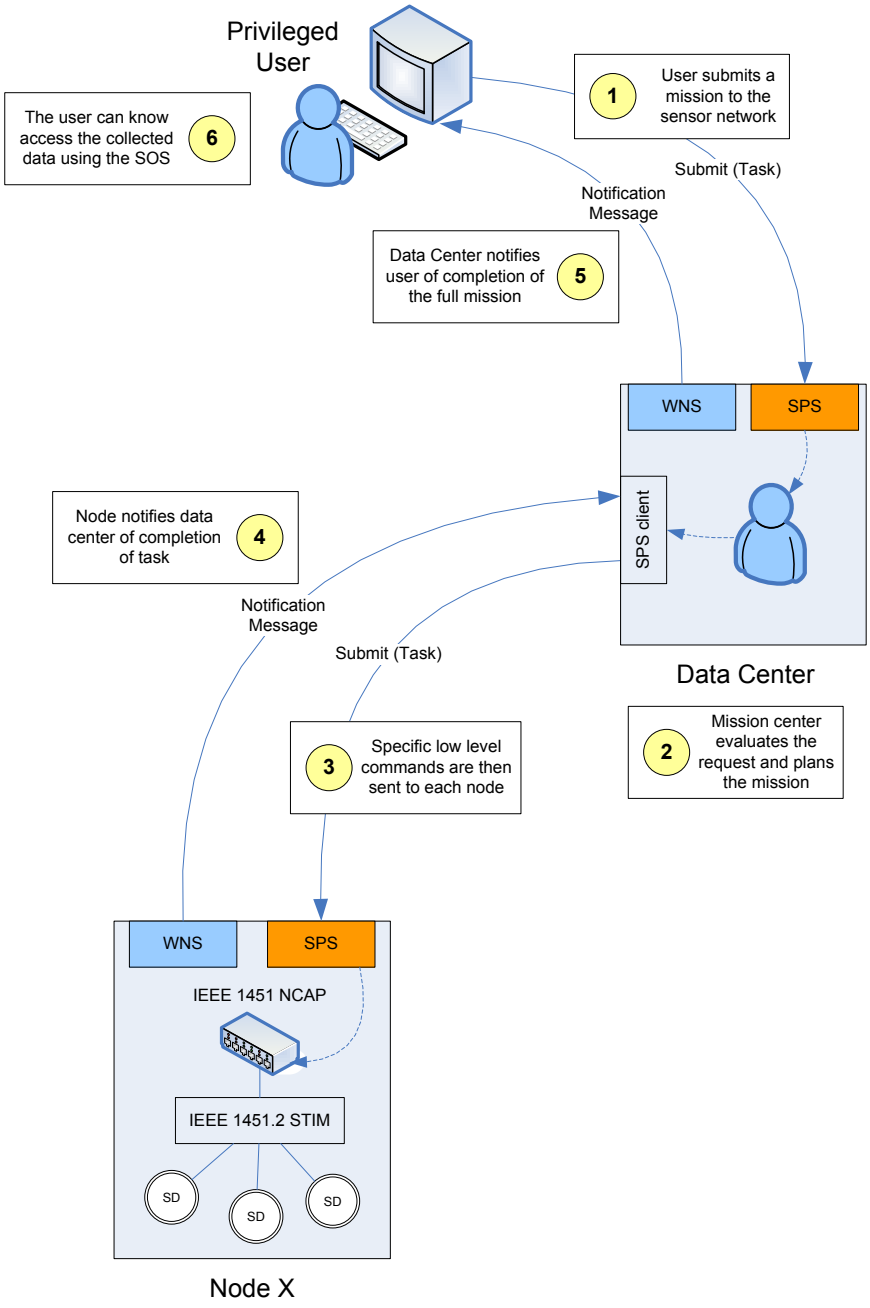
provided contact information using the WNS and subscribed to the alert using the SAS.



Note that if the data center **always** listens to **all** alerts coming from this node, step 1 is not necessary and the notification can be issued systematically by the node software. However, adding a real SAS interface to the node provides more flexibility since different network entities will be able to subscribe to different alerts. This could be of interest if the data center uses more than one service for listening to alerts or if nodes need to listen to each other in order to cooperate on a specific measurement mission.

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

This last diagram shows how a hierarchy of delegated SPS can be used to issue commands to the sensor network that range from overall mission instructions at the data center level to detailed commands sent to an individual node. The task submitted at the data center is intended to be high level, whereas tasks submitted to individual nodes are low level (simple commands). This diagram doesn't show the different SPS 'discovery' operations in details (i.e. *DescribeTasking*, *GetFeasibility*, etc…) but rather assumes that these were done prior to this sequence.



Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

10 Future Directions

An ongoing focus of the SWE architects and the SWE WG is to move all SWE component specifications to a level of maturity to warrant approval as OGC Technical Specifications. The table below shows the various levels of approval for the SWE components:

- **Sensor Model Language (SensorML)** – Approved as OGC Standard
- **Observations and Measurements (O&M)** – Approved as OGC Standard
- **Transducer Model Language (TML)** – Approved as OGC Standard
- **Sensor Observation Service (SOS)** – Approved as OGC Standard
- **Sensor Planning Service (SPS)** – Approved as OGC Standard
- **Sensor Alert Service (SAS)** – OGC Best Practice Paper. RFC process finished.
    - Next step: Formation of Standard Working Group (SWG)
    - Approximate release as OGC Standard: 2008-2009
- **Web Notification Service (WNS)** - OGC Best Practice Paper
    - Next step: Formation of Standard Working Group (SWG)
    - Approximate release as OGC Standard: 2008-2009
- **Web Catalog Service: SWE Profiles** – not published

The SensorML and TML specifications introduce the possibility of significant paradigm shifts in the implementation and application of sensors and sensor networks in the future. Many of these paradigm shifts are already envisioned, including (1) the ability to discover, access, and process sensor observations without a priori knowledge of sensor systems, (2) the ability to package expertise and algorithms within an XML instance and to distribute and execute this algorithm anywhere on the web, (3) the ability to distribute processing of observations at any level of the sensor network, from high-powered data centers to the users hand-held computer, and (4) the ability to enable direct transmission of raw sensor observations, in real time, directly to the user with instructions on processing observations to meet that user's specific needs. Other paradigm shifts will become more apparent as more implementations of sensor web components are realized.

The SWE architects have long recognized that simulations and models should be included within the SWE design and can be supported using the existing architecture. Within the SWE framework, simulations and models are, like sensors, simply sources of observations that can be described (in SensorML), with observations can be discovered and accessed (through a SOS), with possible alerts that can be advertised and published

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

(through a SAS), and that can be configured and tasked (though a SPS). Future work should focus on including simulations and models into the SWE framework, as well as enabling the interoperable opportunities between sensors and models.

Currently, two Standard Working Groups are formed: SPS SWG and SWE Common SWG. At this point, we have to refer to the charters of those working groups as well as published change requests, as both SWG haven't had their constitutional meeting (expected for April 2008). An important aspect will be the potential integration/usage/referring of OASIS Standards (from WS-X suite, e.g. WS-Notification and WS-Addressing).

10.1 OWS-6 Potential SWE Directions

- Recommendations for SWE-related topics for the upcoming OWS6 Interoperability Project will be discussed at the OGC TC meeting in St. Louis. We refer to the meeting notes on the OGC Portal.

11 Open Source Implementations of SWE Components

The following overview lists Open Source Implementations of SWE components. This list is not exhaustive!

- 52°North: Implementation of all SWE services, http://www.52north.org
- VAST, University of Alabama in Huntsville: XML Parser, Data Processing API and Specific Implementations for SensorML, http://vast.uah.edu/SensorML
- Mapserver: SOS implementation, http://mapserver.gis.umn.edu/
- Seagis SOS implementation, http://seagis.sourceforge.net/ (supposed to migrate to http://constellation.codehaus.org/ soon)

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

Annex A

Abstract Test Suite

There is no abstract test suite to the SWE Architecture. Abstract test suits are defined for the individual parts of the architecture.

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

Annex B

XML Schema Documents

All schemas are defined as part of the SWE suite standards and specifications.

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

Annex C

UML model

All UML models are provided in previous clauses.

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

Bibliography

[1] SANY Consortium (2008): Specification of the Sensor Service Architecture Version 1. Online at http://www.sany-ip.eu

[2] Richer, Havens (2005): Sensor Fusion. Theory and Application. White Paper

[3] Simonis, I. and Wytzisk, A. (2003): Simulation Models in SDI. Integrating Simulation Standards in Geo-processing. GIM-International 10 (17) pp. 56-59

[4] Simonis, I. (2007) The Sensor Web: GEOSS's Foundation Layer. In: GEO: The Big Picture. Tudor Rose

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.