GINISSENSE - Applying OGC Sensor Web Enablement

Natasa Veljkovic, Sanja Bogdanovic-Dinic, Leonid Stoimenov
Faculty of Electronic Engineering, University of Nis

INTRODUCTION

In recent years, the Sensor Web concept has been significantly explored and many of its applications related to environmental protection have been proposed. Open Geospatial Consortium (OGC) has certainly had the most important role in promoting Sensor Web. The OGC Sensor Web Enablement working group has published a set of specifications named *Sensor Web Enablement (SWE)* (Simonis, 2008), which fully describe all key components that a Sensor Web based system should implement. The group is constantly working on further improvements of SWE, enabling companies which are working on development of the proposed services, to register and publish their achievements on the OGC Web site.

The Sensor Web system implies collecting real time data from heterogeneous and distributed sensor networks and delivering it to the intelligent system components where it is correlated in order to produce new knowledge that can help make new decisions in the future. Sensor technology has significantly improved during the last few years. Sensors are smaller, more reliable, energy efficient, applicable in various environments and also less expensive. They can be placed anywhere and can measure any type of data, making it easier to monitor the environment and to respond in time when a threat appears. Thus, delivering measured values to the operator on time is very important. SWE proposes Sensor Alert Service (SAS) as a service responsible for collecting data from sensors and delivering it to the subscribed users, and Web Notification Service (WNS) as a service responsible for delivering notifications to the registered users. Together, these two services make an excellent alert-delivering system. This paper deals with implementation of these two services and their application in monitoring and notifying users about sensor events within the GINISSENSE system (Markovic et al., 2009).

The rest of the paper is organized as follows: The second chapter refers to related work. The third chapter is about the GINISSENSE SWE architecture. The fourth chapter describes SAS and WNS, followed by the description of their implementation given in the fifth chapter. The last chapter describes implementation of the GINISSENSE SWE Client.

RELATED WORK

There are several groups currently working towards developing a Sensor Web framework for delivery of physical sensed data. Microsoft Research group has developed the SenseWeb system (Santanche et al., 2006) which provides common platform and set of tools for data owners to easily publish their data and users to make useful queries over the live data sources. GeoSWIFT (Liang et al., 2004) is a framework capable of communicating with webcams as its sensing medium and does include data storage capabilities in the form of registry service. Besides these two, there are many other groups, whose research is based on OGC SWE specifications.

The official OGC Web site enables registration and review of companies working on implementing or that have already implemented some of the SWE services. Figure 1 shows comparative review of registered SAS and WNS implementations, emphasizing some important characteristics. As it can be seen, currently, there are only two complete implementations of SAS (1st Spatial Group and 52°North) and one implementation of WNS (52°North), while other implementations are still in progress. Precisely for these reasons, there is no detailed information about the work of these two companies.

Company Organization	Version	Technology	Status
1st spatial Group	WNS 0.0.9, SAS 0.9.0	JAVA	Implementing, Finished
52° North	WNS 0.0.9, SAS 0.9.0	JAVA	Finished
SAIC	WNS 0.0.9, SAS 0.9.0	.NET	Implementing

Figure 1: Registered SAS and WNS.

While registered, Web services of these companies (presented in figure 1) haven't yet received the status *compliant*, which indicates that implemented Web services comply with the given versions of the specification.

The CG&GIS Laboratory at the Faculty of Electronic Engineering in Nis has been exploring the field of Sensor Web for several years and has developed GinisSense architecture for crisis management system (Markovic and Stoimenov, 2009). The system is designed to work with heterogeneous data sources and follows the OGC SWE specifications.

GinisSense SWE Architecture

The GinisSense SWE Architecture (as illustrated in figure 2) consists of five layers. The architecture is described in details in (Markovic et al., 2009).

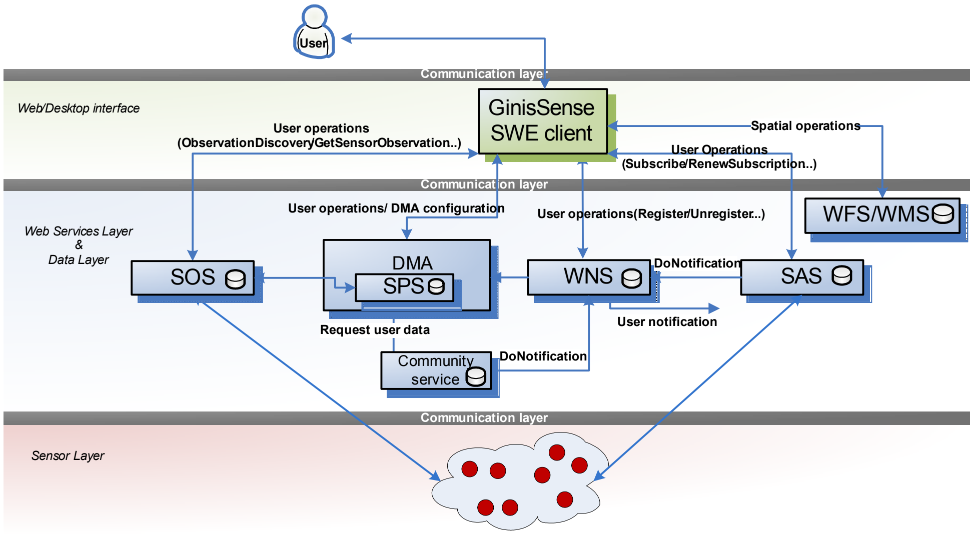


Figure 2: GinisSense SWE architecture.

Sensor layer. Sensors are hardware devices that produce a measurable response to changes in physical conditions of objects. Sensors measure physical parameters of an object or area of interest. Each sensor has an appropriate area which it covers, and the sensor can reliably and accurately report about it. Characteristics and requirements for sensors are to be small, to have low power consumption, to be independent and adapted to the surroundings.

Data layer. This layer is used to store various types of data required by the system. This primarily includes real time sensor data, spatial data necessary to display the sensor position and objects of interest on the map, as well as data collected by users who contribute with gathered information regarding objects of interest. For each data type, there is a separate database used for data storing.

Web Services Layer. Web services layer is responsible for planning, acquisition, analysis and notification of sensor-measurements. There are seven Web services under this layer. All of them, except Community service, are internal services, the specifications of which comply with OGC SWE specifications. Web services are described in detail in (Markovic et al., 2009). SAS and WNS implementations within the GinisSense system will be described in the following section.

Web/Desktop interface. To access information, users are required to use an appropriate Web/Desktop interface. Web interface can be imagined as a WebGIS client (Bogdanovic et al., 2008). Besides the basic GIS functionality, the Web/Desktop GIS application provides users with support for accessing Web services layer. Spatial location, from which the sensor information is obtained, is very important in the analysis, which is why GIS is used as default. A prerequisite for GIS is needed, since it is important to create geo-information infrastructure for sharing and collecting data. In this paper we will present the Desktop GIS interface solution.

Communication layer. This layer controls the sending of data and commands between sensor layer and the Web services layer, as well as between users and Web services. This layer includes media, protocols, topology, etc. The possible communication layers are the Internet, satellite, mobile-phone or radio networks.

SENSOR ALERT AND WEB NOTIFICATION SERVICES

Sensor Alert Service

The SAS is used for registration of sensors and subscription of users for the exchange of sensor data. The SAS provides operations to register sensors (data producers) at the service application as well as operations to subscribe users (data consumers) for available observations. The SAS performs filtering of sensor data based upon the filter criteria defined by the users during subscription. How the sensor data is being sent, is defined by the user during subscription for sensor data. During subscription, the user can choose sensors and filters according to which sensor data will be queried. Filters can be sensor location or measured value. Whenever matches are discovered, a notification is sent to the subscriber, using asynchronous, push-based communication mechanisms. Sensor measurements are being sent as alerts using XML based protocol - XMPP (Extensible Messaging and Presence Protocol) or WNS. Users subscribed for particular sensor information (metadata) or measured value, will receive alerts - notifications of events of their subscription.

Web Notification Service

The WNS enables asynchronous message interchanges between clients and other SWE services. This service is especially useful when many collaborating services are required to satisfy a client‘s request, and/or when significant delays are involved in satisfying the request.

There are two types of communications between the user and WNS: *one-way* and *two-way* communication. One-way communication involves only sending notifications to users, without expecting their replies. Two-way communication requires users to send back asynchronous responses to the received notifications. One-way communication is typically used for sending alerts by SAS, or any other notices that do not require any additional user actions. But there are situations when the user is expected to provide some additional information necessary for further processing and analyzes of the received request. In those cases, two-way communication is used, meaning that the notification is sent to the user and the user must send back his reply. The communication between the user and the service can take place via various different communication protocols: XMPP, HTTP, Email, SMS, Phone and Fax. Based on the GINISSENSE system requirements, suitable protocols are implemented.

WNS collaborates with at least two other SWE services: Sensor Planning Service (SPS) and Sensor Alert Service (SAS). SPS directly communicates with the end users and accepts users' requests. It is responsible for planning and executing those requests, while the responses are being sent via WNS. SAS is responsible for data sources and user subscriptions. One of the tasks of this service is delivering notices to subscribed users about sensor events. Those notices are usually delivered over XMPP, but in some cases the user wants to receive alerts in some other way (email, sms, phone or fax). SAS then uses WNS without having any knowledge about transport protocols that will be used for the delivery of alerts.

GINISSENSE SAS AND WNS IMPLEMENTATION

GinisSense SAS and WNS Web services have been developed using .NET C# and XML technologies and Visual Studio 2008 framework. This choice of technology has been directly influenced by the need of our laboratory for this system, as well as previous experience in developing similar systems. .NET represents a set of technologies focused on the Windows platform and has an excellent built-in support for efficient and easy development of XML-based Web services.

GinisSense WNS has been developed in accordance with the OGC specification (Simonis and Echterhoff, 2007b) while SAS with OGC specification (Simonis and Echterhoff, 2007a). Both specifications imply that communication with service operations is to be performed by exchanging XML messages. Specifications include XML schemas that fully describe the structure and content of XML communication messages.

Each Web services operation-call, first has to be validated against XML schemas, and only in case of successful validation can be sent for further processing. Each operation response is also an XML document which has to comply with the appropriate schemas. For more efficient and easier work with XML and validation of XML requests and responses, we used Visual Studio tool named XML Schema definition tool (XSD.exe). XSD.exe is a console tool which, among other roles, provides generating common language runtime classes from XSD files. That practically means that each XML schema is mapped into exactly one corresponding class, which can be instantiated and treated like any other object. .NET framework also has an excellent built-in support for validating XML documents against concrete set of schemas. XSD.exe tool was used for mapping XML schemas to C# classes, while validation of the incoming XML requests and outgoing XML responses is completely performed using .NET built-in classes.

```
POST /WebService/WebService.asmx/OperationName HTTP/1.1
Host: HostName
Content-Type: application/x-www-form-urlencoded
Content-Length: length

request=string
```

Figure 3: Example of HTTP POST request.

Operations are accessed via HTTP, by sending XML formatted requests. The request must comply with the proposed schemas, which is particularly important while developing a client that will use this service. The operation response also has to be in accordance with XML schemas. The request can be sent either directly, through the Web interface of the Web service, or indirectly, from a client application or other Web service. The HTTP GET request can be sent only for *GetCapabilities* operations. The HTTP POST request is possible for all operations (as shown in figure 3).

WebService (from figure 3), needs to be replaced with GinisSense Web service name, e.g. WNS or SAS. *OperationName* is the name of the calling operation e.g. GetCapabilities. *HostName* is the IP address of the server which hosts GinisSense Web services, e.g. 160.99.9.131. *Request* parameter is the xml document.

GinisSense WNS Implementation

GINISSENSE WNS implementation involves all required and optional *Update* operations (as shown in figure 4).

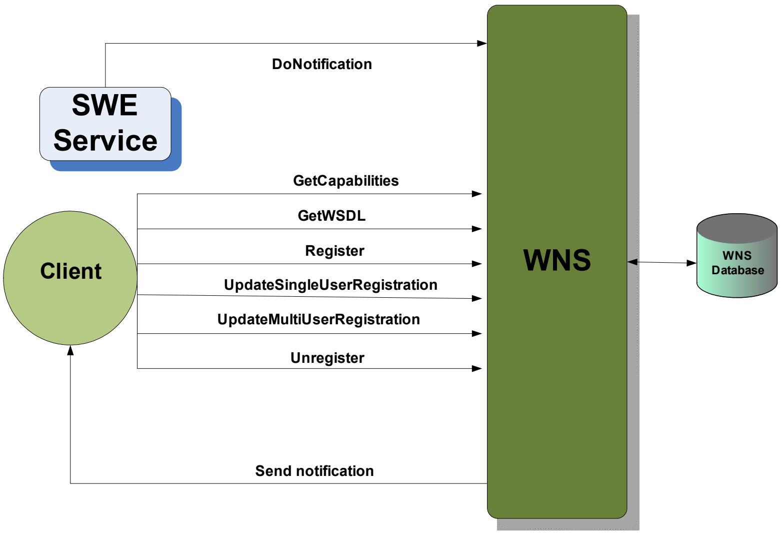


Figure 4: Implemented WNS operations.

The implemented version of WNS allows users to register for receiving notifications. It is possible to register a new, single user, or a group of already registered users. In the second case, notifications are sent to all members of the group. Each user can withdraw his registration at any time, while all information about him will be deleted. The service also allows updating registrations of single and multi users. A single user can update the name and add/remove communications protocols, while multi user can add/remove members to/from the group. Notifications are sent asynchronously to registered users over chosen and supported communication protocols.

This version doesn't contain implementations of all proposed operations and doesn't support all communication protocols, but that is going to be done in the future development of GINISSENSE system within our laboratory.

GinisSense SAS Implementation

GINISSENSE SAS implementation includes all mandatory and optional operations (as shown in figure 5).

All operations are clearly divided to producer and consumer operations. The SAS offers the capability for producers (usually sensors) to advertise offers and to renew or cancel an advertisement. Consumers (usually human users or other applications) can use the Subscribe operation to subscribe for certain alerts. Before a user can subscribe for sensor measurements, he must be aware of sensors that are advertising at SAS. He can learn this by performing the *GetCapabilities* request.

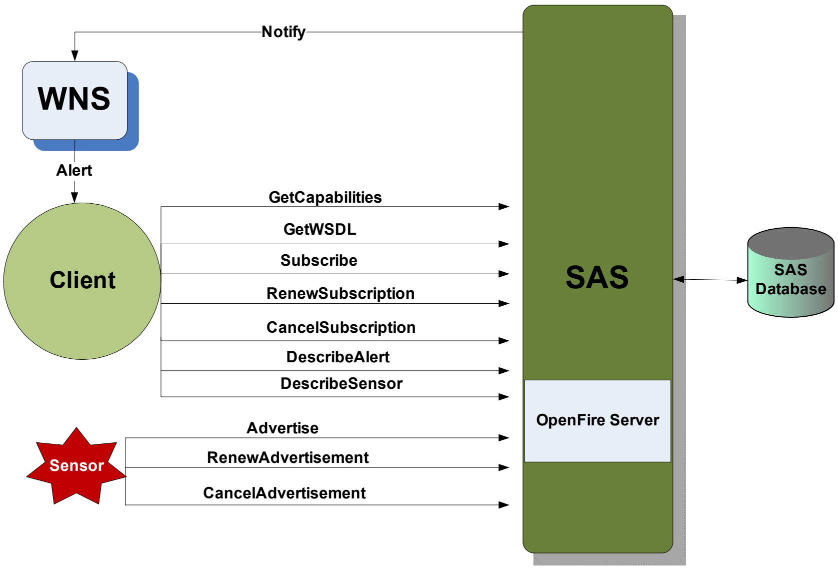


Figure 5: Implemented SAS operations.

The response of *Advertise* operation contains SensorID and Multi User Chat (MUC). The SAS instructs the Openfire server (XMPP Messaging Server) to create a new MUC, in case that sensor has not advertised the same offer before. Otherwise the SAS will simply provide an existing MUC. On the basis of advertised sensors, users can subscribe to self-defined alert conditions using the *Subscribe* operation.

When the conditions are matched, the SAS uses WNS for user notification. Specification (Simonis and Echterhoff, 2007a) proposes push-based delivery of alerts via XMPP or WNS. In the GinisSense SAS implementation, for now, only delivery via WNS is supported.

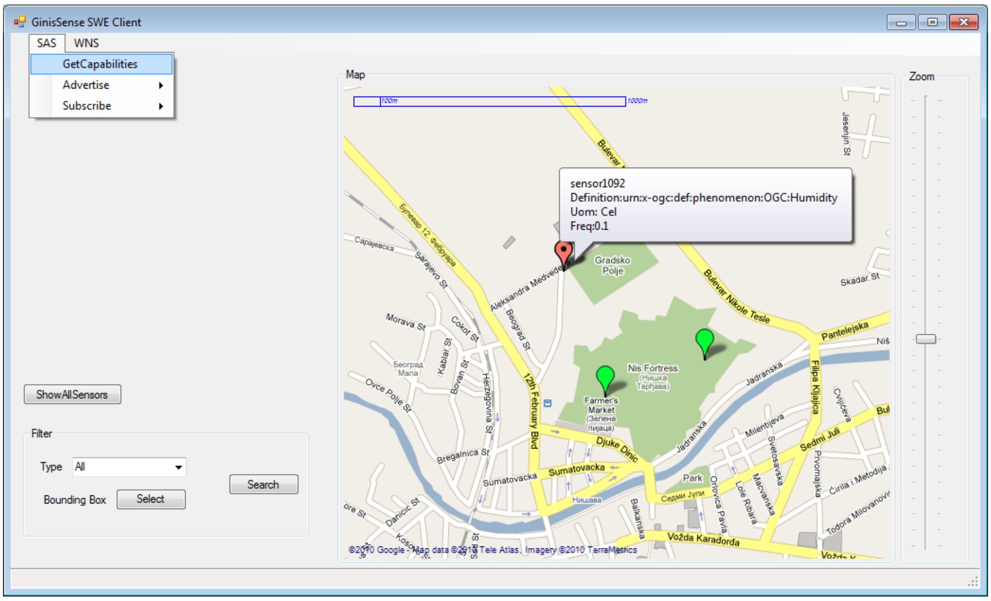
The implemented WNS and SAS are hosted on the laboratory's Server and can be accessed at the address: http://wns.geonis.rs and http://sas.geonis.rs respectively.

GINISSENSE SWE CLIENT

During the past few months, our laboratory has been focusing on research and development of SWE services, SAS and WNS, and their integration and collaboration, as described in previous chapters. For purposes of further research we have developed a desktop client application which uses those implemented services and allows users to register and access real time sensor data.

The application allows registering new sensors and publishing sensor observations through SAS, and registering new users and sending SAS alerts through WNS. There are no limitations in type of sensors that can be registered, giving this client a general dimension and enabling easy adaptation for any type of system. Each user can access the map showing locations and basic information about each registered sensor or sensor system.

The map is interactive, allowing users to query sensors and get information about features of observable phenomena, general information (location, periods of measurements etc.) and status information (whether sensor is on or off at the moment) (as shown in figure 6).



sensor1092
Definition:urn:x-ogc:def:phenomenon:OGC:Humidity
Uom: Cel
Freq:0.1

Figure 6: GINISSENSE SWE client.

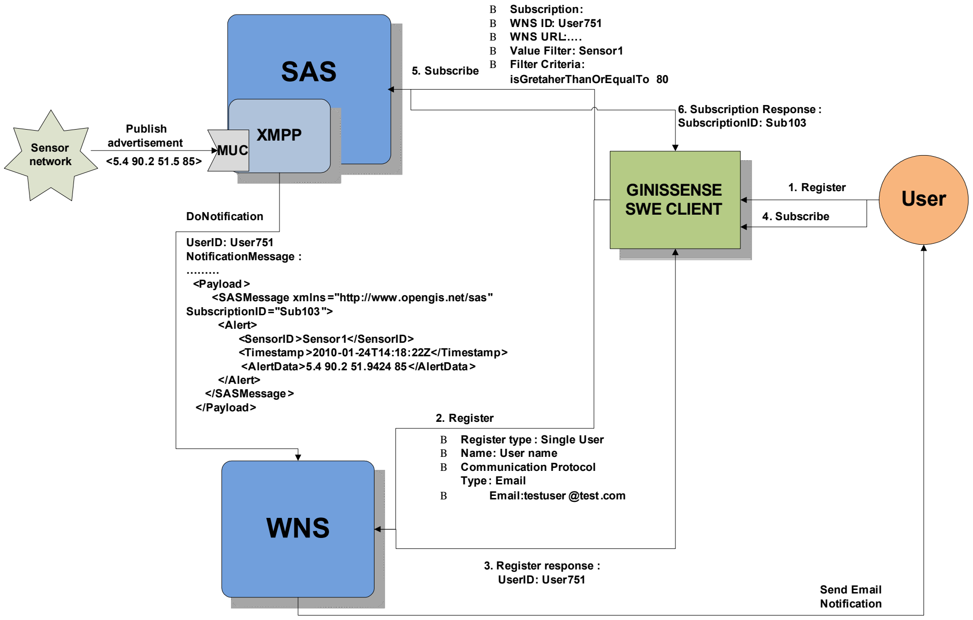
Initially, on starting SWE client, the SAS *GetCapabilities* operation is called in order to receive information about available sensors and their locations and visualize them on the map. Upon the *GetCapabilities* call, the select-list *Type* is filled based on the received information about types of measurements for each available sensor. The same effect can be accomplished by pressing the *ShowAllSensors* button.

Filter panel help users to filter sensors that will be visible on the map. The overview of sensors can be filtered based on measurable parameters (type), or location (bounding box).

A user can select each visible sensor to see detailed information about it. The selected sensor will be painted red, while other available sensors, which haven't been selected, will remain green. The information about the selected sensor will be shown in the info box.

In order to receive alerts about sensor events, the user must first register him/herself with WNS and then subscribe for alerts with SAS. The subscription can be for alerts from specific sensors or for specific types of alert data (for example temperature) within an area. Each registered sensor, in a certain period of time, sends its measurements to SAS via XMPP, which are then stored, processed and sent to all subscribed users over WNS *DoNotification* call (as illustrated in figure 7).

Registration of new sensors is very simple, and information about it is almost immediately available through the client to all users. As soon as sensors make an advertisement, users can subscribe for receiving alerts about it. That way, users are always provided with real time information about sensors and sensor events by the client.



```
UserID: User751
NotificationMessage :
.........
  <Payload>
      <SASMessage xmlns="http://www.opengis.net/sas"
SubscriptionID="Sub103">
        <Alert>
            <SensorID>Sensor1</SensorID>
            <Timestamp>2010-01-24T14:18:22Z</Timestamp>
             <AlertData>5.4 90.2 51.9424 85</AlertData>
        </Alert>
     </SASMessage>
  </Payload>
```

Figure7: General scenario for SWE client

CONCLUSION AND FURTHER WORK

The environmental protection is a noticeable problem and a challenge nowadays, which requires our full attention and effort. Monitoring real time data is of paramount importance for preventing or mitigating disasters. Thanks to the rapid technologies-development, we are now able to design efficient monitoring systems. The Sensor Web has become a standard in this area and is a concept focused on environmental issues and ways of dealing with them.

In this paper, we have presented our contribution to this concept in form of the GINISSENSE system and so far developed services. Our future plans are to continue improving these services and develop SWE SOS and SPS in order to create a general system that can be applied in various disaster management and environmental applications. Special attention will be given to the problems of processing gathered data, its integration, visualization, as well as problems of extracting knowledge from the data obtained from sensors.

BIBLIOGRAPHY

Bogdanovic M., Davidovic N., Antolovic I., Stanimirovic A., Stojanovic D., Stoimenov L., WebGIS application for viewing and analysis of electric power supply network geodata, YUINFO, Kopaonik, Serbia, 2008. (In Serbian)

Liang S.H.L., Tao V., Croitoru A., Sensor Web and GeoSwift - An Open Geospatial Sensing Service, ISPRS XXth Congress, Istanbul, Turkey, pp.1-6., 2004.

Markovic N., Stanimirovic A., Stoimenov L., Sensor Web for River Water Pollution Monitoring and Alert System, 12th AGILE International Conference on Geographic Information Science "Advances in GIScience", Hannover, Germany, ISSN 2073-8013, 2009.

Markovic N., Stoimenov L., Sensor Web Architecture for Crisis Management, Proceedings on XLIV International Scientific Conference on Information, Communication and Energy Systems and Technologies - ICEST 2009, Veliko Trnovo, Bulgaria, Vol. I, pp. 263-266, 2009.

Santanche A., Nath S., Liu J., Priyantha B., and Zhao F., Senseweb: Browsing the physical world in real time, in Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN'06), Nashville (TN), USA, April 2006.

Simonis I., Echterhoff J., Sensor Alert Service Implementation Specification, OpenGIS Best Practise Paper, http://www.opengeospatial.org/standards/bp, OGC Document Number: 06-028r5, Version:0.9.0, 2007a.

Simonis I., Echterhoff J., Web Notification Service Implementation Specification, OpenGIS Best Practise Paper, http://www.opengeospatial.org/standards/bp, OGC Document Number: 06-095, Version:0.0.9, 2007b.

Simonis I., Sensor Web Enablement Architecture, OpenGIS Best Practise Paper, http://www.opengeospatial.org/standards/bp, OGC Document Number: 06-021r4, Version:0.4.0, 2008.