

## Knowledge Bases in Spatial Data Infrastructures – A new Level of Decision Support

Ingo Simonis<sup>1</sup>, Andreas Wytzisk<sup>2</sup>

<sup>1</sup>Institute for Geoinformatics, University of Muenster, Germany

Tel: +49-251-83-33084, Fax: +49-251-83-39763, Email: [simonis@uni-muenster.de](mailto:simonis@uni-muenster.de)

<sup>2</sup>International Institute for Geo-Information Science and Earth Observation (ITC)

Enschede, Netherlands

Email: [wytzisk@itc.nl](mailto:wytzisk@itc.nl)

### ABSTRACT

Spatial Data Infrastructures (SDI) define a framework that allows online access to distributed geographic data and geoprocessing capabilities. They are typically based on distributed web services and aim to be a complementary approach to the predominating local management and processing of data sets. SDIs develop their real performance on national or international scales only in case that cross-system and cross-institutional interoperability is achieved. International accepted standards help to realize such large-scale distributed infrastructures assembled by numerous heterogeneous, but interoperating services. One of the most promising standardization efforts, which provides a suitable foundation to establish a standardized spatial data infrastructure is represented by the OpenGIS consortium (OGC). Beginning with a discussion of the OGC view and realisation of the Reference Model for Open Distributed Processing (RM-ODP) this article gives an in-detail view of this approach. Starting from OGC's "Sensor Web Enablement" initiative, which allows the integration of real time data into SDI, an alerting framework is proposed, that make use of formalized knowledge which is applied to spatiotemporal measurement data provided by an SDI environment.

**KEYWORDS:** Spatial Data Infrastructures, Rules Based Systems, Sensor Web, Interoperability

### Introduction

The standardization work of the Open GIS Consortium is currently state of the art within the geospatial domain. Based on the ISO/IEC RM-ODP, it has produced mature specifications that provide a high level of interoperability. This paper describes the architectural framework of OGC, discusses the Sensor Web Enablement initiative and will end with a proposal how formalized knowledge about facts and circumstances might be integrated into standards-based SDI. The integration will force a break with the traditional stateless and work-upon-request schema that predominates the overall architecture currently.

### Spatial Data Infrastructures

The web service concept represents the next generation architecture that provides a high level of interoperability between software components based on software industry standards. "Web Services can be defined as loosely coupled, reusable software components that semantically encapsulate discrete functionality and are distributed and programmatically accessible over standard Internet protocols" (Sleeper 2001) goes the

# Global Spatial Data Infrastructure

February 2-6, 2004 Bangalore, India

7<sup>th</sup> Conference



long definition trying to explain web services. A more programmatic approach is provided by Vasudevan (Vasudevan 2001): "They are self-contained, self-describing, modular applications that can be published, located, and invoked across the Web". What is making web services so interesting for the GIS community? The answer lies in the monolithic approach that underlies traditional GIS. Loose coupling techniques are used to achieve interactions between different GIS; mainly based on a common file format used for data exchange (Nyerges 1992). This file format is usually not capable to fulfil all the needs of the exchanging systems. Finally, it leads to information loss. The solution would be a completely integrated system existing on top of a common data and method base, but this solution was hard to find (Bernard and Krüger 2000).

The Open GIS consortium (OGC) represents the most promising approach to establish such an integrated system. Founded in 1994 as a non profit organization, OGC has the mission "to deliver spatial interface specifications that are openly available for global use" (OGC 2002). It has currently more than 260 members, including public and private companies, universities, government agencies and other organizations from all over the world who are interested in building spatial interoperability. Thus OGC is one of the world leading organizations addressing interoperability issues within the spatial domain.

OGC's reference model is mainly based on the ISO/IEC Reference Model for Open Distributed Processing (RM-ODP). This reference model creates an architectural framework; support of distribution, interacting, and portability can be integrated (ISO/IEC 1996) into it. This framework comprises of five viewpoints, called enterprise, information, computational, engineering, and technology viewpoint which provide a basis for the specification of ODP systems. Further it describes a viewpoint language for each viewpoint, specifications of the functions required to support ODP systems, and transparency prescriptions showing how to use the ODP functions to achieve distribution transparency.

From the five viewpoints, only two should be discussed in this paper: First and shortly the enterprise viewpoint to give an overview over the currently developed services used within business processes. Second, the computational viewpoint which shows the central role of web services within the OGC approach to achieve an interoperable system; hence it is of major interest for this paper. To keep this article to a sensible length, only specific parts of the particular viewpoints are picked out. The other viewpoints mainly deal with information models and schemas required by an ODP application to transform data (information viewpoint), and the communication architecture which supports heterogeneous distributed computing, portability and interaction between applications (engineering viewpoint). The technology viewpoint deals with specific hard- and software components. Those viewpoints are out of focus of this paper.

## Enterprise Viewpoint

The short introduction to the enterprise viewpoint which is given here, is limited to an overview of the different components and their use within the OGC approach. This paper does not capture all user requirements on the infrastructure necessary to support new and existing services and the business practices of the entire environment.

The OpenGIS Web Services Framework provides a common set of interfaces and encodings that span the functional parts of the enterprise. The OGC web services based on this set could be subdivided into five types:

# Global Spatial Data Infrastructure

February 2-6, 2004 Bangalore, India

7<sup>th</sup> Conference



1. **Registry services:** services that provide a common mechanism to classify, register, describe, search, maintain, and access information about network resources (data and services). Currently Web Registry Service (WRS) and Web Catalog Service are the two services of this type.
2. **Processing-workflow services:** Services that act as intermediate services. They negotiate between web service environment users and geospatial resource providers or bridge between those components that are not able to follow a synchronous request-response model. Web Notification Service (WNS) and Sensor Planning Service (SPS) belong to this category.
3. **Portrayal services:** Services that provide specialized capabilities to support the rendering process of spatial data. Web Map Service (WMS), Web Coverage Service (WCS), and Style Management Service (SMS) constitute this category currently.
4. **Data services:** Services that provide geospatial data, using a standardized XML encoding. Web Object Service (WOS), Web Feature Service (WFS), Sensor Collection Service (SCS), Image Archive Service (IAS), and Web Coverage Service (WCS) belong to this type.
5. **Data processing services:** Services that allow the processing of standardized encoded geospatial data. The Web Coordination Transformation Service (WCTS) is the only service of this type.

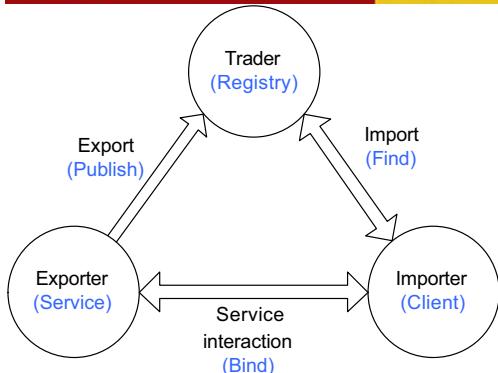
Interacting services, integrated into a common business application, form a spatial data infrastructure (SDI). Until 2002, OGC service developments focused mainly on the interoperability of data accessing and visualization capabilities and the representation and interchange of geographic data. As there are a couple of service specifications available supporting the interchange and use of static geographic data, the business demand to access spatio-temporal data and related operations grows. Therefore, first actions have been taken by standardizing service interfaces for the so called "Sensor Web Enablement". We will get to this bridge between static geospatial data and time dependent information later in this article.

## Computational Viewpoint

The Computational Viewpoint takes the functional interface description from the Informational Viewpoint and adds the elements, e.g. data types that allow the applications to be computable. At this point, we will turn the attention to three aspects: Firstly the service oriented architecture (SOA) pattern, the interface structure, and finally the used encodings.

## SOA pattern

From a computational point of view, OGC's approach is based on the service oriented architecture (SOA) pattern. Basically, this architecture comprises components, services, and processes. Components are binaries that do specific tasks and have a well defined interface. A service is simply a grouping of components whereas processes join services into a business. This architecture combines the ability to invoke remote objects and functions with tools for dynamic service discovery, placing an emphasis on interoperability. One of the fundamental concepts that addresses discovery and use of services is the Publish-Find-Bind pattern, elaborated primarily in ISO/IEC (13235-1) and redefined by the Object Management Group (OMG 2000). Figure 1 illustrates the pattern, showing OMG expressions in black, OGC correspondent roles in grey.



**Figure 1: Publish-Find-Bind Pattern**

In the RM-ODP framework, there are three fundamental roles:

- **Trader** - a role which registers service offers from exporter objects and returns service offers upon request to importer objects according to some criteria (OGC expression is “registry”, though a trader is a service as well).
- **Exporter** - a role which registers service offers with the trader object (OGC expression is “service”, though any kind of self describing service).
- **Importer** - a role which obtains service offers, satisfying some criteria, from the trader object (OGC expression “client”, though it might be a service as well).

If a service provider (exporter role) wants to export/publish a service offer, it gives the trader a description of the service to offer. This description includes a description of the interface at which that service instance is available. A client seeking a service has to ask the trader for that service by providing a service description containing the necessary service characteristics. The trader checks the registered service descriptions and – if a positive match was found – responds to the importer with the necessary information required to bind with a service instance.

### Interfaces

The interfaces of the SOA components – means the service interfaces – are organized in a hierarchical structure. Common sets are integrated into top level interfaces. All interfaces do specialize an abstract WSInterface interface, add new operations to it as well as overload existing ones.

### Encodings

To achieve a high level of interoperability, the individual components need a common data model and encoding to transfer data. For this reason, all transferred data is XML encoded, using standardized schemas. Currently more than six different XML grammars, all written in XML Schema for the modeling, transport, rendering, and storage of geographic information exist. Three of these grammars are of major interest for this paper:

1. **Geographic Markup Language (GML)**: Used for the modelling, transport, and storage of geographic information. GML provides a variety of kinds of objects for describing geography including features, coordinate reference systems, geometry, topology, time, units of measure and generalized values (Cox et al. 2003).

2. Sensor Model Language (SensorML): SensorML provides a XML Schema for defining the geometric, dynamic, and observational characteristics of a sensor (Botts 2002).
3. Observation and Measurement (O&M): O&M is an encoding framework for measurements and observations (Cox 2002) and since 2003 part of GML.

### SWE – Sensor Web Enablement

To bridge the aforementioned gap between static geospatial data and time dependent information, a first starting point is found in the “Sensor Web Enablement” (SWE). The initial focus in SWE has been to investigate standardised interfaces for live sensors operating in near-real-time, rather than the conventional static data stores. Note, however, that a simulator resembles a sensor regarding the provision of spatio-temporal data (it only differs in the way how it estimates the requested value and in being temporally independent). SWE addresses information gathering from distributed, heterogeneous, dynamic information sensors and sources of different structure, based on web services.

It is the goal to develop common access, planning, and management interfaces and a descriptive markup language (SensorML) for managing sensor information and metadata in common consistent manners, independent of any application.

The individual parts were initially designed to fulfill the following needs (Simonis et al. 2003):

- Describe sensors in a standardized way
- Standardize the access to observed data
- Standardize the process of what is commonly known as sensor planning, but in fact is consisting of the different stages planning, scheduling, tasking, collection, and processing
- Building a framework and encoding for measurements and observations

Point one, the standardized description of sensors is realized using SensorML, which represents not only a mean to describe the characteristics of a sensor, but those of the mounting platform in addition. To standardize the access to observations provided by sensors or by a sensor networks, the interface of the SensorCollectionService was developed. The SensorPlanningService is a web service used for organizing observations of natural phenomenon. The framework and encoding for measurements and observations is realized in the XML grammar Observations&Measurements. During SWE development it became obvious that against the background of handling long duration processes a further service is needed: A notification service that sends notifications to users in case that the observation operation could be performed or that an observation is done, delayed or even cancelled. The WebNotificationService provides the capabilities (Simonis and Wytsisk 2003). It is the first asynchronous service and therefore presents a change in the traditional OGC paradigm to stick on synchronous services only. In the following, the five pillars of the Sensor Web Enablement are described shortly.

### Sensor Collection Service

The Sensor Collection Service aggregates readings from live, in-situ sensors. The service provides an interface to get O&M-encoded data, information about the sensor itself (encoded in a SensorML instance document), and information about the sensor mounting platform (SensorML as well).

# Global Spatial Data Infrastructure

February 2-6, 2004 Bangalore, India

7<sup>th</sup> Conference



## Sensor Planning Service

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 support systems that surrounds them (Lansing 2002). A SPS not only has to support different kinds of assets with differing capabilities, 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 resulting observation data. The SPS is designed to be flexible enough to handle such a wide variety of configurations.

## Web Notification Service

Like mentioned before, the Web Notification Service breaks with the paradigm of synchronous service handling. This became necessary because the period between starting a task (e.g. an observation performed by a sensor) and its completion is undefined. A mechanism became necessary to register users at a centralized location and make them addressable from this service. The service interface allows beyond pure user registration the sending of two different kinds of messages. The well defined message schema is therefore divided into two subparts: A notification message and a communication message. The first one is used to notify the user that an event has occurred without expecting any kind of answer from the user. In case that an event occurs that requires user consultation or further authorization (e.g. an observation could not be performed with the required quality), the second message type is sent, a communication message. This underlying message schema is more complex than that of a simple notification message and allows the mapping of user replies to individual application threads. Currently the WNS supports four communication channels: e-mail, SMS, Instant Message, and HTTP POST (which allows triggering of further web services) (Simonis and Wytsisk 2003).

## SensorML

Based on common understanding among earth observation scientists that data from space-borne sensors is not adequately nor easily georeferenced to meet the requirements, a markup language was developed that allows the description of the dynamic, geometric, observational characteristics of sensors. This common basis allows a much easier mapping and recalculation of the data provided by different space-borne sensors. During the three years lasting development phase, SensorML was extended to provide support for a large range of sensors and is not restricted to space-borne sensors only. In its current shape it (Botts 2002):

- provides general sensor information in support of data discovery
- supports the processing and analysis of the sensor measurements
- supports the geolocation of the measured data.
- provides performance characteristics (e.g. accuracy, threshold, etc.)
- archives fundamental properties and assumptions regarding sensor

SensorML, which is an XML Schema, provide functional models for sensors, not a detailed description for the hardware itself. It treats sensors and sensor platforms separately.

# Global Spatial Data Infrastructure

February 2-6, 2004 Bangalore, India

7<sup>th</sup> Conference



## Observation and Measurement

The Observations and Measurements Engineering Specification describes a conceptual model that allows the depiction of spatial and temporal variant readings which values are the results of the estimation of some natural phenomenon. The representation is based on the specified XML encoding. For Cox, measurement usually refers to the measuring device and procedure used to determine the value, such as a sensor or observer, analytical procedures, simulations or other numerical processes. The carrying out of the procedure to estimate the value of the phenomenon is called observation (Cox 2002).

## Knowledge Integration

At present, knowledge about facts or circumstances has to be hard coded in proprietary applications that in fact may use of standards-based SDI to obtain the necessary data. Focusing the current status of the Sensor Web Initiative, a knowledge formalizing strategy would gain huge benefit in matching service provision and real application design.

Imagine the following scenario: As a lorry fleet manager, you would like to be informed in case that a smog alarm level is announced, because it forces you to direct your lorries on different routes which bypass the restricted area. If you know all facts about smog, you would not need to wait for the news speaker within the local radio, but simply make use of a web based sensor network, interconnected by a SDI. All you would have to do is to register your smog rules at a specialised web service. The service would screen web based catalog services for matching sensor collection services. In a next step, this service would poll the current measurement values at a predefined interval and will inform you in case that all your rules are matched. We designed the Web Alert Service do master this business.

## Web Alert Service – WAS

Primarily, WAS will be part of the Sensor Web and responsible for initiating the dispatch of notifications in case predefined rules are satisfied. These rules have a rather abstract nature which means, that links to existing information resources are established not until the rule registration has taken place. These linked resources will be SCS in particular, but also conventional OGC Web Services like Web Feature Services (WFS) or Web Coverage Services (WCS). The latter is the case if a rule evaluation requires non-measurement data like political boundaries for example.

Against the background of typical alerting use cases, which often require a fast provision of measurement information, it seems advisable to reassess OGC's recent web service architecture as a suitable platform for the Sensor Web. Focusing exclusively on data pulling, the service trading (publish-find-bind) based approach shows significant shortcomings, if an active event dispatch respectively data push is required, which is the case if a measurement value has to be immediately distributed if it exceeds a predefined threshold.

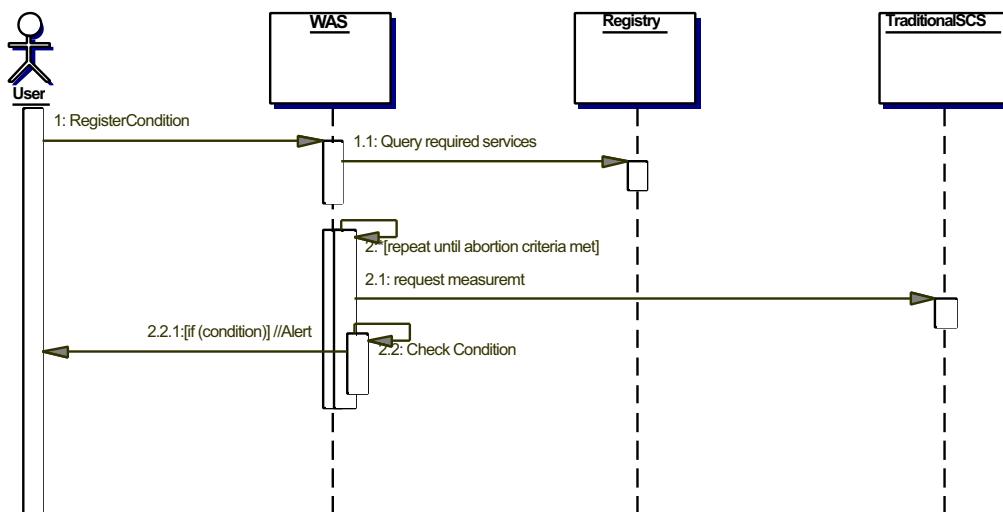
Software architectures for event generation, observation and notification, which support loose coupling and heterogeneity among components are well developed (see particularly Rosenblum & Wolf, 1997) and form – from an architectural point view – a complementary approach to the existing web service architecture. Therefore, in the following sections a synergetic SWE approach is proposed, which extends the

traditional service-trading architecture by event-based patterns to support both, pull- and push-based data provision mechanisms.

### Pull model

In the traditional OGC service model, data is synchronously provided by a service upon request, means that the data is pulled from the providing service, which has to be located by using a registry service. In the SWE framework, this would mean that the WAS would have to pull the observation results periodically, namely in the same way it has to access temporally more static data provided by a WFS or WCS. Figure 2 illustrates this in an exemplary manner. A typical use case for the pull model is the following:

WAS supervises a water depths sensor within a river system. The sensor measures the depths every 30min. A user might be interested in the observed values only once an hour. In its capabilities, the SCS specifies its observation intervals. WAS would pull the current value only once an hour. No changes to the SCS would be necessary. Due to the fact that it is very unlikely for the water depths to change erratically, it is sufficient to provide the data pulling functionality.



**Figure 2: Pull Model**

### Push model

The pushing model follows the event observation and notification model described in Rosenblum and Wolf, 1997 [4]. The framework is constituted by objects<sup>1</sup> which can be assigned to the following roles:

- objects of interest are objects of which operations are invoked and that are possibly sending an event if an operation is invoked
- invokers are objects that invoke an operation of an object of interest; an invoker possibly sends an event if it calls an object's operation

<sup>1</sup> like processors, storage devices, network devices, or some other hardware components of the network, as well as any logical entity residing on a hardware component, such as a file, a program, a process or a communication packet

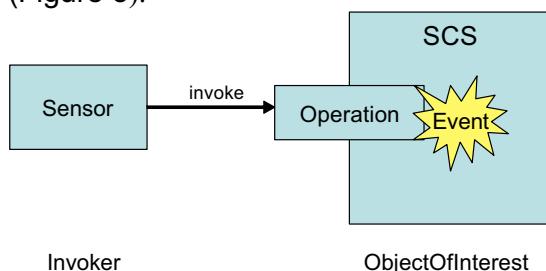
- observers are detecting occurrences of events and sending notifications to recipients, which showed interest in the detected events
- recipients are entities that receive notifications about events.

The interaction of event sending objects (objects of interest, invokers), observers and recipients follows a publish/subscribe metaphor: objects that send events publish information about the types of events which are intended to be dispatched, recipients subscribe in those event type occurrences they want to be informed about, and observers send notifications to recipients if they detect an event occurrence a recipient has subscribed for.

In the SWE framework sensors play the role of invokers, SCS represent objects of interests, and observers are represented by a combination of WAS and Registry Services. This combination represents a major difference to conventional event based systems: Usually, the observer, as a single entity, is constituted by two parts: Firstly as being the target of the “publication registrations”, means that objects register their event types at this entity, and secondly as being the target of “sent events”, means that objects send their events to this very same entity. Theoretically, this would be possible within the SWE framework as well, but it would mean to combine registry and WAS functionality in a single service. Even if it would facilitate things under certain conditions, we do not propose to dump OGC registries. Registries and WAS should coexist in the SWE framework. WAS scan the registries for specific event types and send a subscription request to the corresponding SCS. Users and/or services that have to be notified in case that an alert condition is matched act as recipients.

### Event Generation

A sensor that just has performed its measurement invokes an operation of the SCS to inform it about the new measurement. It could be an update operation in case that the sensor measures always the same observable at the same location with the same configuration but with a different timestamp, or a newObservation operation if any of the further parameters had changed. The termination of this operation invocation generates an event which can be seen as the instantaneous effect of the (normal or abnormal) termination of an invocation of an update or newObservation operation (Figure 3).



**Figure 3: Event Generation**

Any event is uniquely described by an O&M fragment (with the parameters sensorID, observableID, observationTime, observationLocation, SCS\_ID).

If any other objects will be notified about the occurrence of the event is dependent on the subscribed event type. Event subscription at the SCS will be performed by the WAS. WAS define simple rules that always follow the syntax

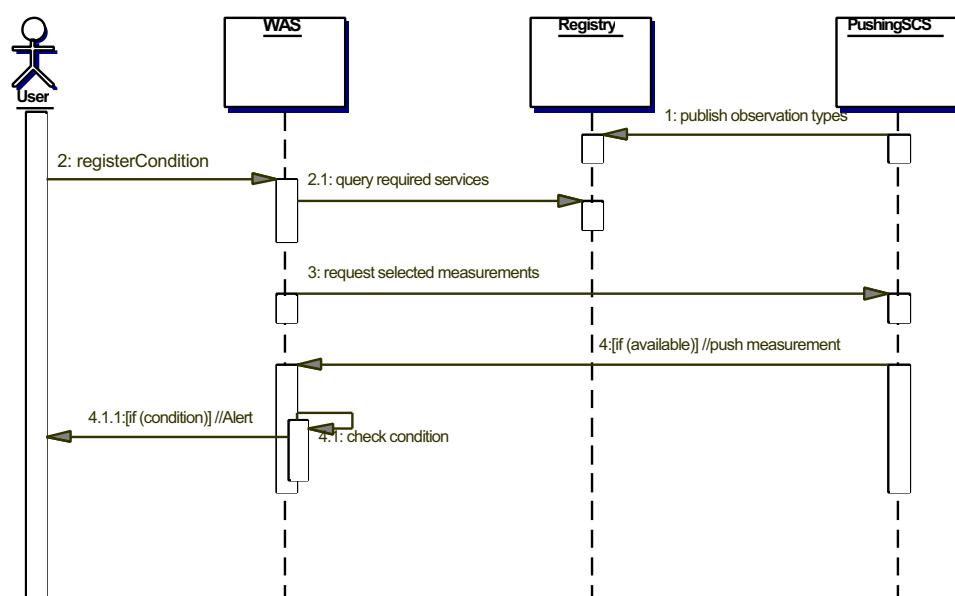
```
<...>
<ObservableID>CO</ObservableID>
<Comparator>IsGreater Than</Comparator>
<Value>240</Value>
<WASCallback>http://w.a.s/callback/interface</WASCallback>
<...>
```

Therefore, SCS will notify about the occurrence of an event only in case that at least one WAS is interested in the event WAS provide a callback interface which will be used to send the event to the WAS.

### The Observer Role of aWAS

The WAS acts as an observer, whereas the SCS decides which of its events become observable. The SCS has to be enriched in its functionality to support the pushing schema. Its interface has to be extended by at least one operation to register the threshold rules. Simultaneously, it has to be equipped with pushing capabilities to be able to push selected events to its observing WAS and to push its capabilities to a registry. The observing WAS queries registry services to find suitable SCS and subsequently sends its "informMeInCaseThat"-requests to the SCS. It is up to the WAS ensure that the units of measurements used by the SCS are matching the threshold value specified in the request. WAS will analyse received events, run its inference mechanisms and use the Web Notification Service to finally inform interested parties or to start any arbitrary process chain. A typical use case with attached sequence diagram (Figure 4) will illustrate this:

WAS supervises a sensor for radioactivity. The observed values (hopefully) will never change. If the value changes, a user wants to be notified immediately. A pulling mechanism would consume unnecessary bandwidth due to the fact that the current observed value has to be requested every second in case of a change in contrast to expectations. In this case, a pushing mechanism is imperative.



**Figure 4: Push Model**

## Conclusion

To support decision making processes and alerting activities as best and as fast as possible it has to be guaranteed that highly temporal dynamic spatial information can easily be linked with additional information resources provided by a SDI. To ensure a fast and flexible notification of those concerned (decision makers, inhabitants within a specific area, emergency management forces, public authorities etc.) interoperable alerting services have to be supplied additionally. Therefore, the presented WAS-framework allows the registration of specific conditions and will double check the corresponding services for accordance. To overcome pure individual sensor limits, WAS is based on a rule based approach to allow even complex multi-sensor based threshold value definitions. As a next step, a prototypical implementation of the proposed concept will proof its suitability for supporting real world scenarios.

## Bibliography

- Bernard, L. and Krüger, T. (2000) Integration of GIS and Spatio-Temporal Simulation Models: Interoperable Components for Different Simulation Strategies. *Transactions in GIS* 4(3): pp. 197-215.
- Botts, M. (2002) Sensor Model Language (SensorML) for In-situ and Remote Sensors DIPR Version 0.7. Open GIS Consortium (OGC), Technical Report OGC 02-026r4.
- Cox, S. (2002) Observations and Measurements DIPR Version 1.2.0. Open GIS Consortium (OGC), Technical Report OGC 02-027r3.
- Cox, S., et al. (2003) OpenGIS® Geography Markup Language (GML) Implementation Specification. Open GIS Consortium, Technical Report OGC 02-023r4.
- ISO/IEC (1996) Information technology - Open Distributed Processing - Reference Model: Architecture. ISO/IEC, Technical Report ISO/IEC 10746-3:1996(E).
- Lansing, J. (2002) Sensor Planning Service DIPR Version 0.0.5. Open GIS Consortium (OGC), Technical Report
- Nyerges, T. L. (1992) Coupling GIS and Analytical Models. Bresnahan, P., Corwin, E., and Cowen, D. (Ed.), Proceedings of: 5th International Symposium on Spatial Data Handling, Charleston, USA, pp. 524-533.
- OGC (2002) OGC Vision, Mission, Goals, Values. Open GIS Consortium, Inc., Technical Report
- OMG (2000) Trading Object Service Specification. OMG, Technical Report
- Simonis, I. and Wytzisk, A. (2003) Web Notification Service DIPR Version 0.1. OpenGIS Consortium (OGC), Technical Report
- Simonis, I., et al. (2003) Interoperability of Simulation and Geoinformation Standards. Proceedings of: Spring-SIW, Orlando, Florida,
- Sleeper, B. (2001) Defining Web Services.  
[http://www.stencilgroup.com/ideas\\_scope\\_200106wsdefined.htm](http://www.stencilgroup.com/ideas_scope_200106wsdefined.htm)

# Global Spatial Data Infrastructure

February 2-6, 2004 Bangalore, India

7<sup>th</sup> Conference



Indian Space Research  
Organisation



Dept. of Science  
and Technology



Global Spatial Data  
Infrastructure

Vasudevan, V. (2001) A Web Service Primer. accessed: 22.02.2003.

<http://webservices.xml.com/pub/a/ws/2001/04/04/webservices/index.html>

C:\Dokumente und Einstellungen\Simonis.IFGI\Eigene Dateien\Literatur\v\Vasudevan 2001 A Web Service Primer.pdf