THE SENSOR WEB TESTBED FOR FLOOD MONITORING AND PREDICTION

Nataliia Kussul, Andrii Shelestov, Sergii Skakun

Abstract: In this paper we provide an overview of emerging Sensor Web paradigm and show several practical issues of using Sensor Web technologies for real-world tasks. Issues under study include sensor description using SensorML and database performance for serving observations data. This paper also shows an approach for integrating standard Sensor Observation Service with Globus Toolkit Grid platform.

Keywords: Sensor Web, Grid, OGC.

ACM Classification Keywords: D.2.12 [Software Engineering] Interoperability; Information Systems; H.1.1 [Models and Principles] Systems and Information Theory; H.3.5 [Information Storage and Retrieval] Online Information Services; I.4.8 [Image Processing and Computer Vision] Scene Analysis - Sensor Fusion.

Conference: The paper is selected from XVth International Conference "Knowledge-Dialogue-Solution" KDS 2009, Varna, Bulgaria, June-July 2009

Sensor Web Paradigm

The Sensor Web is an emerging paradigm and technology stack for integration of heterogeneous sensors into common informational infrastructure [Mandl et al., 2006; Moe et al., 2008]. The basic functionality required from such infrastructure is remote data access with filtering capabilities, sensors discovery and triggering of events by sensors conditions.

The Sensor Web is governed by a set of standards developed by the Open Geospatial Consortium [Botts et al., 2007]. At present, the following standards are available and approved by consortium:

— OGC Observations & Measurements (http://www.opengeospatial.org/standards/om) – Common terms and definition for Sensor Web domain;

— Sensor Model Language (http://www.opengeospatial.org/standards/sensorml) – XML-based language for describing different kinds of sensors;

— Transducer Model Language (http://www.opengeospatial.org/standards/tml) – XML-based language for describing the response characteristics of a transducer;

— Sensor Observations Service (http://www.opengeospatial.org/standards/sos) – an interface for providing remote access to sensors data;

— Sensor Planning Service (http://www.opengeospatial.org/standards/sps) – an interface for submitting tasks to sensors.

There are also standards drafts that are available from Sensor Web working group but not yet approved as official OpenGIS standards:

— Sensor Alert Service – service for triggering different kinds of events basing of sensors data;

— Web Notification Services – notification framework for sensor events.

Sensor Web paradigm assumes that sensors could belong to different organizations with different access policies or, in broader sense, to different administrative domains. However, existing standards stack does not provide any means for enforcing data access policies leaving it to underlying technologies. One possible way for handling informational security issues in Sensor Web is presented in the next subsections.

Sensor Web Flood Use Case

One of the most challenging problems for the Sensor Web technology implementation is a global ecological monitoring in the framework of the Global Earth Observation System of Systems (GEOSS). Decision makers in an emergency response situation (e.g. floods, droughts) need to have a rapid access to the existing data sets, the ability to request and process data including the specific of emergency, and tools to rapidly integrate the various information sources into a basis for decisions. In this paper we consider the problem of flood monitoring using satellite remote sensing data, in-situ data and results of simulations.

The flood monitoring and prediction scenario presented here is being implemented within the GEOSS AIP (Architecture Implementation Pilot, http://www.ogcnetwork.net/AIpilot). It uses precipitation data from the Global Forecasting System (GFS) model and NASA's Tropical Rainfall Measuring Mission (TRMM, http://trmm.gsfc.nasa.gov) to identify the potential flooded areas. Once the areas have been identified, we can request satellite data for the specific territory for flood assessment. These data can be both optical (like EO-1, MODIS, SPOT etc) and microwave (Envisat, ERS-2, ALOS, Radarsat-1 etc).

The problem of floods monitoring by itself consumes data from many heterogeneous data sources such as remote sensing satellites (we are using data of ASAR, MODIS and MERIS sensors), in-situ observations (water levels, temperature, humidity, etc). Floods prediction is adding the complexity of physical simulation to the task.

The Sensor Web perspective of this test case is depicted in Fig. 11. It shows collaboration of different OpenGIS specifications of Sensor Web. The data from different sources (numerical models, remote sensing, in-situ observations) is accessed through Sensor Observation Service (SOS). Aggregator site is running Sensor Alert Service to notify interested organization of possible flood event using different communication mean. Aggregator site is also sending orders to satellite receiving facilities using Sensor Planning Service (SPS) to get satellite imagery only available by preliminary order.

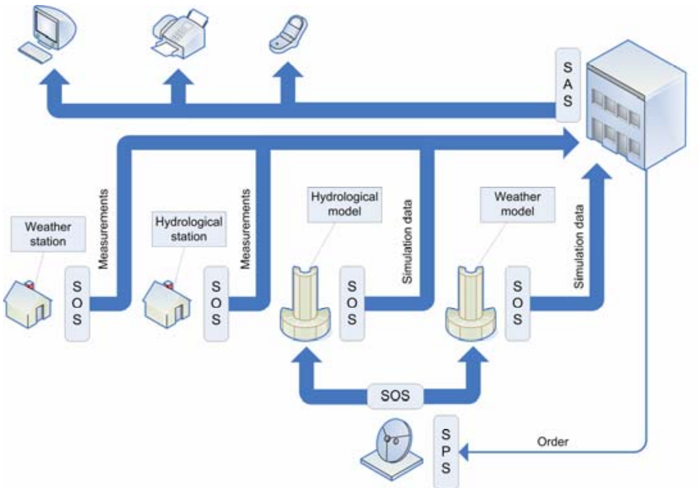

Figure 1. Sensor Web testbed for flood monitoring and prediction

SensorML Description of NWP Model

Sensor Modeling Language (SensorML) is the cornerstone of all Sensor Web services. It provides a comprehensive description of sensor parameters and capabilities as well as sensor calibration lineage, measure errors characteristics, response curves and other information about sensor. SensorML can be used for describing different kind of sensors:

- Stationary or dynamic;
- Remote or in-situ;

- Physical measurements or simulations.

Modelling and simulation are very important parts of environmental monitoring. The importance of different models in the process of solving of real-world tasks was demonstrated in the previous part of this paper. Sensor Web infrastructure should be able to integrate modelling data and provide remote data access for the as well as other Sensor Web features like discovery, sending orders, etc.

At the bare minimum, SensorML description should contain general information about sensor (time and geographical extents, contact persons, etc) and lists of inputs and outputs. SensorML input could be either physical phenomena or some external measured value. The first case applies to physical measuring devices and second – to models and simulations.

We have tried to describe weather modelling process using WRF [Kussul et al., 2009] numerical model in terms of SensorML. There are nearly 50 inputs and 20 outputs for basic WRF configuration. It's obvious that information density of inputs and outputs descriptions in SensorML is quite low and each of them requires quite significant amount of XML code to be properly described. The problem lies in very verbose description of multidimensional data. Three- and four- dimensional data arrays are very common in environmental modelling but SensorML provides poor experience regarding them.

Authors have raised this problem during thematic meeting and hope that next revision of SensorML will include some elements for simpler description of multidimensional data.

Sensor Observation Service Implementation

In order to provide access to hydrometeorological observations over the regions of interest we have deployed Sensor Observation Service implementation on the site of Space Research Institute of NASU-NSAU. We have studied two possible implementations of SOS for particular task of serving temperature sensors data. Implementations under study were:

- UMN Mapserver v5 (http://mapserver.gis.umn.edu/)
- 52North SOS (http://52north.org/)

The advantages and disadvantages of these solutions can be summarized in the following table.

	UMN Mapserver v5	52North SOS
Advantages	1. Very good and reliable abstraction for different data sources (raster files, spatial databases, WFS, etc) 2. Simple application model (CGI executable) 3. Wide set of features beside SOS 4. Open software	1. SOS implementation is stable and complete 2. Platform-independent (Java-based) 3. A part of wider Sensor Web implementations stack (SPS, SAS) 4. Open software 5. Source code is clean and easily reusable
Disadvantages	1. SOS support is declared but far from being working implementation 2. Poor documentation on SOS topic 3. Strange plans for future development (in particular, automatic SensorML generation)	1. No data abstraction: the only data source is relational database of specific structure 2. Database structure is far from optimal (strings as primary keys, missed indexes, etc) 3. Complex application model (Java web application)

The best experience received was with 52North SOS server. Its main disadvantage is complex relational database scheme. However it was possible to adapt existing database structure to the one, required by 52North using a number of SQL views and synthetic tables. The details of database adaptation are given in the next section.

We have used 52North implementation for building a testbed SOS server providing data of temperature sensors over Ukraine and South Africa regions. The server is available by URL http://web.ikd.kiev.ua:8080/52nsos/sos.

SOS output comes as XML document in special scheme, specified by SOS reference document. The standard is describing two possible forms of results, namely “Measurement” and “Observation”. The first form is more suitable to the situations when the service is returning small amounts of heterogeneous data. The second form is most suitable for long time series of homogeneous data. The table below provides an example of SOS output in these two forms and clearly shows the difference.

Measurement

```
<om:Measurement gml:id="o255136">
 <om:samplingTime>
  <TimeInstant xsi:type="gml:TimeInstantType">
   <timePosition>
    2005-04-14T04:00:00+04
   </timePosition>
  </TimeInstant>
 </om:samplingTime>
 <om:procedure xlink:href=
  "urn:ogc:object:feature:Sensor:WMO:33506"/>
 <om:observedProperty xlink:href=
   "urn:ogc:def:phenomenon:OGC:temperature"/>
 <om:featureOfInterest>
  <sa:Station gml:id="33506">
   <name>WMO33506</name>
   <sa:sampledFeature xlink:href=""/>
   <sa:position>
    <Point>
     <pos srsName="urn:crs:epsg:4326">
      34.55 49.6
     </pos>
    </Point>
   </sa:position>
  </sa:Station>
 </om:featureOfInterest>
 <om:result uom="celsius">10.9</om:result>
</om:Measurement>
```

Observation

```
<om:result>
2005-03-14T21:00:00+03,33506,-5@@
2005-03-15T00:00:00+03,33506,-5.2@@
2005-03-15T03:00:00+03,33506,-5.5@@
2005-03-15T06:00:00+03,33506,-4.6@@
2005-03-15T09:00:00+03,33506,-2.2@@
2005-03-15T12:00:00+03,33506,1.7@@
2005-03-15T15:00:00+03,33506,1.7@@
2005-03-15T18:00:00+03,33506,2.4@@
2005-03-15T21:00:00+03,33506,-0.7@@
2005-03-16T00:00:00+03,33506,-1.4@@
2005-03-16T03:00:00+03,33506,-1.1@@
2005-03-16T06:00:00+03,33506,-1.1@@
2005-03-16T09:00:00+03,33506,-1.3@@
2005-03-16T12:00:00+03,33506,0.5@@
2005-03-16T15:00:00+03,33506,1.7@@
2005-03-16T18:00:00+03,33506,1.5@@
</om:result>
```

Database Issues

The database of hydrometerological information of Space Research Institute of NASU-NSAU contains nearly 1.5 millions of records with observations started at year 2005 to the present moment. The data is stored in PostgreSQL database with PostGIS spatial extensions. Most of the data records are contained in single table 'observations' with indexes built over fields with observation time and station identifier. Tables of such volume requires some special handling so the index for time field was clusterized thus reordering data on the disks and reducing the need for I/O operations. Clusterization of time index reduced typical queries times from 8000 ms to 250 ms.

To adapt this database to the requirements of 52North we have created a number of auxiliary tables with reference values related to SOS (such as phenomena names, sensor names, regions parameters, etc) and a set of views that transforms underlying database structure into 52North scheme. 52North's database scheme uses string primary keys for auxiliary tables instead of synthetic numerical and is far from optimal in sense of performance. It doesn't have strong impact on performance with record counts in these tables less than one hundred but will surely cause problems in large-scale SOS-enabled data warehouses.

The typical SQL query from 52North service is quite complex (see listing below). An average response time for such query (assuming one month time period) is about 250 ms with PostgreSQL running in virtual environment on 4 CPUs server with 8GB of RAM and 5 SCSI 10k rpm disks in RAID5 array. Increasing of query depth results in linear increasing of response time with estimate speed of 50 ms per month (see Fig. 2).

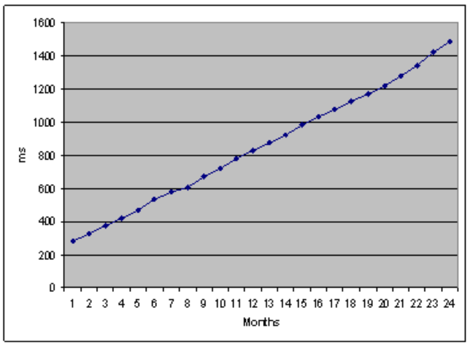

Fig. 2. Dependency between depth of query and response time

Sensor Web SOS Gridification

Sensor Web services like SOS, SPS and SAS, can benefit from the integration with the Grid platform [Foster, 2002; Fusco et al., 2007; Shelestov et al., 2006] like Globus Toolkit (htpp://www.globus.org). Many Sensor Web features can take advantage of the Grid platform services, namely:

— Sensors discovery could be performed through the combination of Index Service and Trigger Service;

— High-level access to XML description of the sensors and services could be made through queries to the Index Service;

— Grid platform provides a convenient way for the implementation of notifications and event triggering using corresponding platform components [Humphrey et al., 2005];

— Reliable File Transfer (RFT) service [Allcock et al. 2005] provides reliable data transfer for large volumes of data;

— Globus Security Infrastructure [Welch et al., 2003] provides enforcement of data and services access policies in a very flexible way allowing implementation of desired security policy.

We have developed a testbed SOS Service using Globus Toolkit as a platform. Currently, this service works as a proxy translating and redirecting user requests to the standard HTTP SOS server (see Fig. 12). The current version uses client-side libraries for interacting with the SOS provided by the 52North in their OX-Framework. The next version will also include in-service implementation of SOS-server functionality.

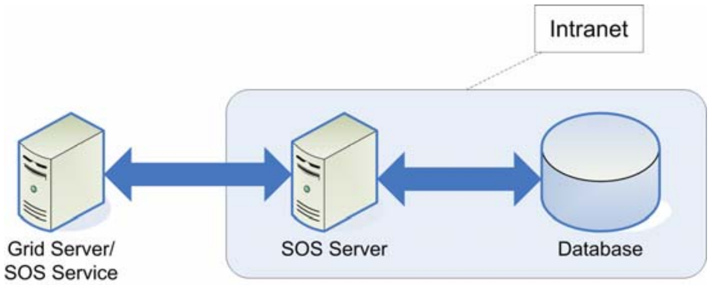

Fig. 3. Grid-based SOS service implementation

Grid service implementing SOS provides the interface specified in the SOS reference document. The key difference between the standard interfaces and Grid-based implementations of the SOS lies in the encoding of service requests. The standard implementation uses custom serialization for the requests and responses, and the Grid-based implementation uses standard SOAP encoding.

To get advantage of the most Globus features, the SOS service should export service capabilities and sensor descriptions as WSRF resource properties (Foster 2005). Traditionally, the implementation of such properties requires translation between XML Schema and Java code. However, the XML Schema of the SOS and related standards, in particular GML (Humphrey et al. 2005), is a very complex one, and there are no available program tools able to generate Java classes from it. We have solved this problem by storing service capabilities and sensor descriptions data as DOM Element objects and using custom serialization for this class provided by the Axis framework that is used by the Globus Toolkit. Using this approach, we can not access particular elements of the XML document in object-oriented style. However, the SOS Grid service is acting as proxy between user and SOS implementation, so it does not need to modify XML directly. With resource properties defined in this way, we can access it using standard Globus API or command line utilities.

Conclusions

Despite of immaturity of Sensor Web technology stack it can provide good experience in serving heterogeneous data of in-situ observations. SOS implementation for serving geospatial raster data that is important for remote sensing data are yet to be implemented. SensorML descriptions of complex environmental models are too verbose. To allow wide use of models in Sensor Web environment some changes should be made in SensorML to shorten descriptions of multidimensional inputs and outputs. Integration with Globus Toolkit Grid platform allows Sensor Web service to take advantage of robust information management features of Grids as well as mature mechanisms for data access policy enforcement.

Acknowledgements

This work is supported by the joint project of the Science & Technology Center in Ukraine (STCU) and the National Academy of Sciences of Ukraine (NASU), “Grid Technologies for Multi-Source Data Integration” (No. 4928).

Bibliography

[Allcock et al., 2005] W. Allcock, J. Bresnahan, R. Kettimuthu, M. Link. The Globus Striped GridFTP Framework and Server. In Proc ACM/IEEE SC 2005 Conf on Supercomputing, 2005.

[Botts et al., 2007] M. Botts, G. Percivall, C. Reed, J. Davidson OGC Sensor Web Enablement: Overview and High Level Architecture (OGC 07-165), 2007, http://portal.opengeospatial.org/files/?artifact_id=25562.

[Foster] I. Foster. The Grid: A New Infrastructure for 21st Century Science. Physics Today, 55(2), pp. 42-47, 2002.

[Fusco et al., 2007] L. Fusco, R. Cossu, C. Retscher. Open Grid Services for Envisat and Earth Observation Applications. In: Plaza AJ, Chang C-I (ed) High performance computing in remote sensing, 1st edn. Taylor & Francis Group, New York, pp. 237-280, 2007.

[Kussul et al., 2009] N. Kussul, A. Shelestov, S. Skakun. Grid and sensor web technologies for environmental monitoring. Earth Science Informatics, 2009, Doi 10.1007/s12145-009-0024-9.

[Humphrey et al., 2005] M. Humphrey, G. Wasson, K. Jackson, J. Boverhof, M. Rodriguez, J. Bester, J. Gawor, S. Lang, I. Foster, S. Meder, S. Pickles, M. McKeown. State and Events for Web Services: A Comparison of Five WS-Resource Framework and WS-Notification Implementations. In: Proc 4th IEEE Int Symp on High Performance Distributed Computing (HPDC-14), Research Triangle Park, NC, 2005.

[Mandl et al., 2006] D. Mandl, S.W. Frye, M.D. Goldberg, S. Habib, S Talabac.Sensor Webs: Where They are Today and What are the Future Needs? In: Proc Second IEEE Workshop on Dependability and Security in Sensor Networks and Systems (DSSNS 2006), pp. 65-70, 2006.

[Moe et al., 2008] K. Moe; S. Smith, G. Prescott, R. Sherwood. Sensor Web Technologies for NASA Earth Science. In: Proc of 2008 IEEE Aerospace Conference, pp. 1-7, 2008.

[Welch et al., 2003] V. Welch, F. Siebenlist, I. Foster, J. Bresnahan, K. Czajkowski, J. Gawor, C. Kesselman, S. Meder, L. Pearlman, S. Tuecke. In: Proc 12th IEEE Int Symp on High Performance Distributed Computing, pp 48-57, 2003.

[Shelestov et al., 2006] A. Shelestov, N. Kussul, S. Skakun. Grid Technologies in Monitoring Systems Based on Satellite Data. J. of Automation and Inf. Sci., 38(3), pp. 69-80, 2006.

Authors' Information

Kussul Nataliia – Deputy Director, Space Research Institute NASU-NSAU, Glushkov Prospekt 40, build. 4/1, Kyiv 03680, Ukraine; e-mail: inform@ikd.kiev.ua

Shelestov Andrii – Senior Scientist, Space Research Institute NASU-NSAU, Glushkov Prospekt 40, build. 4/1, Kyiv 03680, Ukraine; e-mail: inform@ikd.kiev.ua

Skakun Sergii – Senior Scientist, Space Research Institute NASU-NSAU, Glushkov Prospekt 40, build. 4/1, Kyiv 03680, Ukraine; e-mail: serhiy.skakun@ikd.kiev.ua