

Study on Robustness Middleware using Integrating Sensor Observation Service in Sensor Web Enablement

Woo Suk Ahn*, Ki Hyung Kim*, Seung Wha Yoo*

* The graduate School of Ajou University, Suwon, 443-749 Republic of Korea

aptitude@ajou.ac.kr, kkim86@ajou.ac.kr, swyoo@ajou.ac.kr

Abstract— Sensor Observation Service (SOS) in Sensor Web Enablement (SWE) proposed by Open Geospatial Consortium (OGC) collects sensing data through Asset Manager. The data is transformed into Observation & Measurement (O&M) and it is transferred to the client. SOS, dependent on the Asset Manager, needs to be synchronized every time the change occurs to the Asset Manager. Also, the expansion of Sensor Web often causes load concentration. In this paper, we propose new SWE architecture integrating various SOS to control the vendor and the schema of the DBMS. This architecture can minimize modification of SOS due to the change of sensor type. Also, the new component of SWE can manage the load of requests.

Keywords— SWE, SOS, Sensor Web, USN, Sensor Web Portal

I. INTRODUCTION

Many studies of Ubiquitous Sensor Network (USN) work on applying sensor in observation and measurement. The results from such exercises should be provided into an integrated form. However, each sensor system has its own architecture different from each other. Open Geospatial Consortium (OGC) proposed Sensor Web Enablement (SWE) that can merge individual sensor systems.

Sensor Observation Service is the SWE components which send the sensing data to client in Observation & Measurement (O&M) form, the predefined data type, using Extensible Markup Language (XML). The general scenario for in-situ sensors is illustrated in Figure 1, where deployed sensors (S_n) of various types are grouped into several constellations (C_n) that are then accessed through some service (e.g., Sensor Observation Service (SOS)) [1][2].

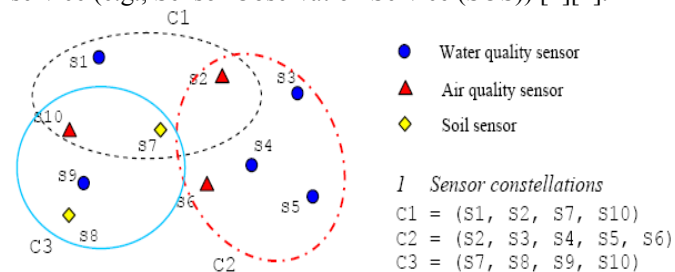


Figure 1. A sample line graph using colors which contrast well both on screen and on a black-and-white hardcopy

As shown in Figure 1, multiple SOSs exist in sensor system. When a historical data is requested, SOS has to retrieve the data from the archive of its own Asset Management System.

Therefore, the information about vendor and the schema of the Asset Manager DBMS should be built by each SOS to answer to the request of historical data. Once the information built, SOS is able to form a static SQL query and send it to DBMS to retrieve historical data.

The retrieving mechanism is dependent on Asset Manager: SOS need to be changed when a new Asset Manager is added or when the DB schema is changed. Under the traditional architecture, the cost of maintenance increases linearly with the accumulation of changes.

The façade SOS (F-SOS) integrates diverse SOS to solve the weakness of the traditional architecture. The F-SOS reduces meaningless processes during the retrieving mechanism by converting the 1:n relationship between client and SOS to 1:1 relationship. Also, it reinforces the robustness of the system by omitting the occurring of publication at every single Asset Manager DBMS vendor and schema changes.

The rest of the paper is organized as follows. In Section II, some works related to the proposed method and their problems is briefly reviewed. Then, we will explain our proposed scheme and case study in Section III, IV. Finally is this paper's future works and conclusion.

II. RELATED WORKS

This Section briefly reviews OGC sensor Web Enablement and Sensor Observation Service because proposed scheme reliant on following that.

A. OGC Sensor Web Enablement

The models, encodings, and services of the SWE architecture enable implementation of interoperable and scalable service-oriented networks of heterogeneous sensor systems and client applications. In much the same way that Hyper Text Markup Language (HTML) and Hypertext Transfer Protocol (HTTP) standards enabled the exchange of any type of information on the Web, the OGC's SWE initiative is focused on developing standards to enable the discovery, exchange, and processing of sensor observations, as well as the tasking of sensor systems.

In general, SWE is the standard developed by OGC that encompasses specifications for interfaces, protocols and encodings that enable discover, access, obtain sensor data as well as sensor-processing services. The following are the seven primary specifications for SWE [2]:

1) Observations & Measurements Schema (O&M): Standard models and XML Schema for encoding observations and measurements from a sensor, both archived and real-time [2][6].

2) 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 of taskable properties [2][7].

3) Transducer Markup Language (TML): The conceptual model and XML Schema for describing transducers and supporting real-time streaming of data to and from sensor systems [2].

4) Sensor Observations Service (SOS): Standard web service interface for requesting, filtering, and retrieving observations and sensor system information. This is the intermediary between a client and an observation repository or near real-time sensor channel [1][2].

5) Sensor Planning Service (SPS): Standard web service interface for requesting user-driven acquisitions and observations. This is the intermediary between a client and a sensor collection management environment [2][3].

6) Sensor Alert Service (SAS): Standard web service interface for publishing and subscribing to alerts from sensors [2][5].

7) Web Notification Services (WNS): Standard web service interface for asynchronous delivery of messages or alerts from SAS and SPS web services and other elements of service workflows [2][4].

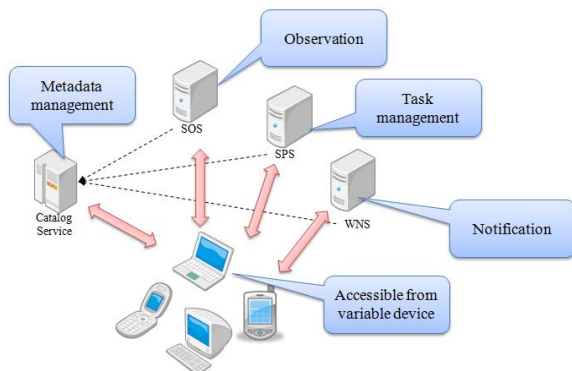


Figure 2. Sensor Web Enablement (SWE) architecture [1]

B. OGC Sensor Observation Service

The OpenGIS Sensor Observation Service Implementation Specification is designed 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. The SOS is the intermediary between a client and an observation repository or near real-time sensor channel. Clients can also access SOS to obtain metadata information that describes the associated sensors, platforms, procedures and other metadata associated with observations [2].

III. PROPOSED ARCHITECTURE

A. SWE Architecture Applying F-SOS Component

We propose a new Sensor Observation System (SOS) component that is named by Façade Sensor Observation System (F-SOS). F-SOS is located in front of SOSs just like a wall. It receive all requests on behalf of SOSs

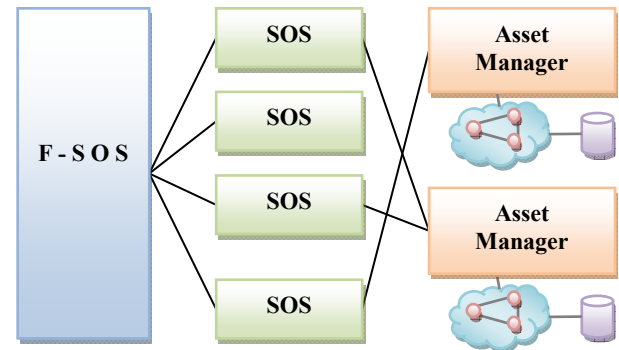


Figure 3. Façade SOS (F-SOS) integrate a number of SOSs.

In general case, a client find SOS associated with Asset Manager to retrieve sensing data that he wants. We suppose that SOS already knows DB schema of Asset Manager. Then, SOS will make a SQL at the request time. The SQL statement will be sent to Asset Manager, and SOS receives sensing data which is the result of requested by client from DBMS of Asset Manager.

In this proposed architecture, F-SOS integrates many scattered SOSs. (We suppose that F-SOS can have unlimited a number of SOSs.) At first, client request F-SOS to gain a sensing data that is collected by sensor. And then, F-SOS tries to find SOS that is associated with Asset Manager. After finding a SOS, it generates a SQL to send to a DBMS of Asset Manager. Finally, F-SOS sends the generated SQL to selected SOS. Unlikely SOS, F-SOS should have DBMS schemas of the entire Asset Manager because the F-SOS provides service directly to client instead of SOSs.

B. Operations of F-SOS

F-SOS is a set of SOSs. Therefore, it comprehends operations of traditional SOS. Mandatory operations in F-SOS are summarized in Table1. It can also use optional operation in OGC Standard [1]:

- RegisterSensor
- InsertObservation
- GetObservationById
- GetResult
- GetFeatureOfInterest
- GetFeatureOfInterestTime

Table 1. Traditional Mandatory Operations of F-SOS

Operation	Description
GetCapabilities (mandatory)	This operation helps retrieving service metadata about a specific service instance for client.
DescribeSensor (mandatory)	This operation helps retrieving detailed sensor metadata.
GetObservation (mandatory)	This operation helps retrieving result of sensing data in O&M.

Table 2. New Operations of F-SOS

Operation	Description
SyncSchema (new)	This operation synchronize DB schema with integrating SOSs.
getData (new)	This operation makes SQL suitable for SOS's DBMS schema.

According to propose new scheme, we need several new operations that is shown Table2.

1) SyncSchema: To manage all of DB schema connected with F-SOS, it always synchronize DB schema with integrating SOSs. We assume that F-SOS has all of data connection sources. F-SOS periodically sends "SyncSchema" operation. And then F-SOS receives each response that is modified DB schema information from SOS that is changed.

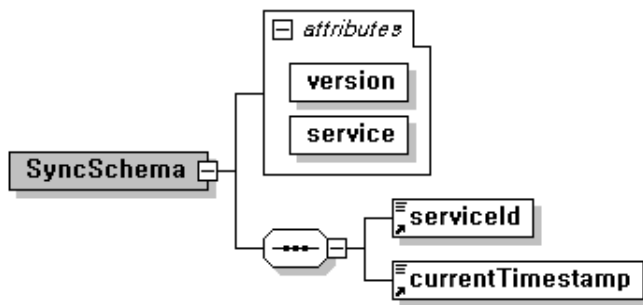


Figure 4. SyncSchema request format.

Figure 4 shows "SyncSchema" operation request message. It contains "serviceld" element and "currentTimestamp" element. The "serviceld" element is identifier of SOS. It can take naming system like a Uniform Resource Names (URN). The "currentTimestamp" element is time of final synchronization. It is useful for testing the system is current.

2) getData: In OGC standard, "getData" operation is a member of SOS component and it retrieve sensing data with pre-defined SQL when it request historical data. However, it is meaningless because it is probable that DB schema of SOSs will be changed. Because, in our proposed scheme, F-SOS has all DB schemes using "SyncSchema", we generate SQL statements dynamically.

C. Benefits of F-SOS

In this paper, we propose F-SOS that integrates scattered SOSs. It has four major benefits for Sensor Web Enablement (SWE):

1) Robustness: We assume that Asset Manager collects sensing data immediately and stores in pre-defined DBMS. Unfortunately, DBMS schema or vendor of Asset Manager is often variable. For instance, when a new type of sensor is admitted into Asset Manager, data type in DBMS needs to be modified because new data type might not be adopted by DBMS. SOSs always should be synchronized with linked Asset Manager. Therefore, every time change occurs to DBMS, SOS has to be refreshed to synchronize with DBMS. In our proposed architecture, it is enough with only one deployment of F-SOS. Reducing meaningless and heavy deployment processes, F-SOS becomes robust to unexpected changes. Although the F-SOS is failure, this system has to service consistently. If the F-SOS is unique in the world, it is impossible to maintain service. Therefore, the F-SOS can be composed of several backup machines. If one of F-SOS is failure, that system is transferred to another machine and will recover quickly.

2) Reducing maintenance cost: F-SOS can reduce maintenance cost. As the number of SOSs is increased, this benefit becomes more important. In the traditional system, the cost increases linear to the changes, because the cost depends on number of SOS. Even if the number of variable SOSs is increased, the cost is kept constant.

3) Load Balancing: If a number of clients request sensing data to SOS directly, then the load can be concentrated in specific SOS. F-SOS can control a load in front of SOSs. First, F-SOS receives the request message from client. It monitors a SOS's resource. It has a scheduling table for managing task. If one of the SOS's resources is full, F-SOS doesn't send request command and store it in own queue. (It can manage with another strategy of scheduling.)

4) Lightness of SOSs: F-SOS saves the works and roles of the SOSs. Without using F-SOS, SOSs have a lot of tasks such as making and querying a SQL statement, and assembling Observation and Measurement (O&M) documents. We propose scheme in which F-SOS makes SQL statement for retrieving a sensing data. Therefore, SOSs simply exist to forward SQL statement received from F-SOS to linked Asset Manager.

5) Caching SQL statement and result: F-SOS can manage all SQL statement because it receives all request messages instead of a number of SOSs. Accordingly, it can cache requests of high frequency and it can provide manufactured data to client using pre-cached information without requesting different messages to SOSs and Asset Manager. We have a lot of study of caching algorithm. In this paper, we introduce only two methods. First, the system caches a SQL statement that is heavily used to reduce load of a manufacturing SQL statement. It is good for limited storage because the statement size is very small. Second, the system stores Observation & Measurement (O&M) that is a result encoded in Extensible Markup Language (XML) about a request of high frequency. If the storage of machine is unlimited, this can reduce response time and save resource for Asset Manager. Thus, there is a trade-off between storage size and performance.

IV. CASE STUDY

We analyse following several case. First, when F-SOS will be built in SWE, we think that F-SOS send synchronizing message to all of SOSs. Second, when F-SOS is failure, how recover the system.

A. Synchronization

Figure 4 shows the sequence of periodical synchronization. First, F-SOS periodically sends "SyncSchema" operation with request message encoded in XML. When SOS receives a synchronization request, it retrieves own DB schema and return response message to F-SOS. Finally, when F-SOS receives response message, it applies its DB schema. F-SOS has to repeat this sequence periodically.

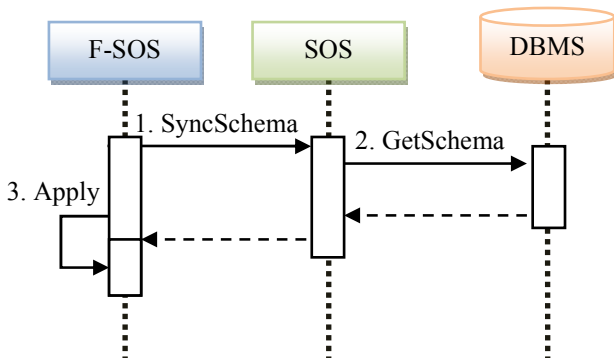


Figure 5. synchronizing DB schema(periodically)

Synchronization period can be changeable. If period value is long, we have no trust in F-SOS DB schema information. And opposite, if we set a short period in this system, F-SOS will be busy and waste resource. Furthermore, if F-SOS has a lot of SOS, it will be down. Figure 6 shows simple sequence of on-demand synchronization. When the F-SOS get request of user for retrieving sensing data, the F-SOS send a command to related SOS. Because the F-SOS synchronize DB schema with SOS on-demand, it is not necessary for periodical update.

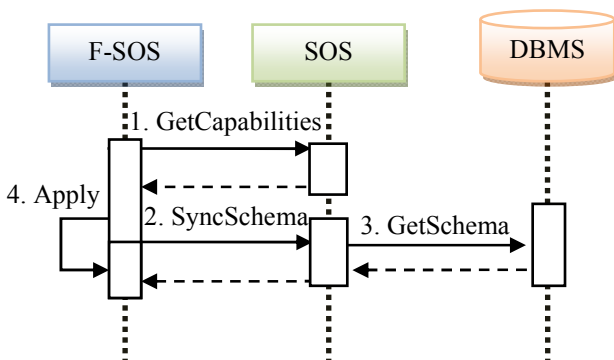


Figure 6. synchronizing DB schema(on-demand)

In traditional system, SOS is turn down and modifies this system when its DBMS schema is changed. Whenever the system is modified, many SOS have to be adapted. However,

the F-SOS is not necessary for waste time because it automatically updates current information.

B. Lightness SOS

The F-SOS is a façade SOS. It is very intelligent because that is a making a decision where will a command send. On the other hand, a traditional SOS only forward simple command and SQL statement. The F-SOS works on behalf of all the SOS (Figure 7).

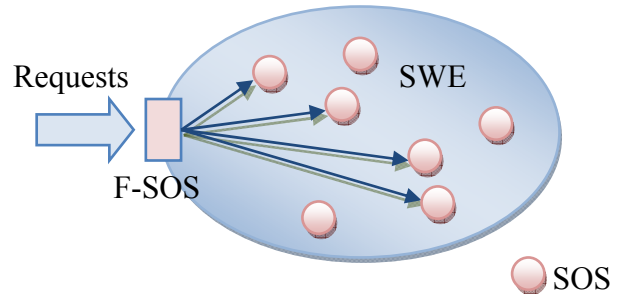


Figure 7. The F-SOS works on behalf of all the SOS.

In our paper, the F-SOS is robustness. However, if that system is crashed, we cannot insist the system is robustness. In this situation, we compose backup system against emergency situation. If one of the F-SOS knocks down, the system occurs rapidly transition.

V. FUTURE WORK & CONCLUSION

We propose the new component in SWE. The F-SOS integrates diverse SOS to solve the weakness of the traditional architecture. It reinforces the robustness of the system by omitting the occurring of publication at every single Asset Manager DBMS vendor and schema changes. It will be enhanced when the SWE is larger than now.

In the future, we hope to develop the system and analyze its performance.

REFERENCES

- [1] Arthur Na, Mark Priest, "Sensor Observation Service", OGC® 06-009r6, Open Geospatial Consortium Inc, 2007
- [2] Mike Botts, George Percivall, Carl Reed, John Davidson, "OGC Sensor Web Enablement: Overview and High Level Architecture", OGC® 07-165, Open Geospatial Consortium Inc, 2007
- [3] Ingo Simonis, University of Muenster, "OpenGIS® Sensor Planning Service Implementation Specification", OGC® 07-014r3, Open Geospatial Consortium Inc, 2007
- [4] Ingo Simonis, Johannes Echterhoff, "Draft OpenGIS® Web Notification Service Implementation Specification", OGC® 06-095, Open Geospatial Consortium Inc, 2006
- [5] Ingo Simonis, "OGC® Sensor Alert Service Candidate Implementation Specification", OGC 06-028r3, Open Geospatial Consortium Inc, 2006
- [6] Simon Cox, "Observations and Measurements – Part 1 - Observation schema", OGC 07-022r1, Open Geospatial Consortium Inc, 2007
- [7] Mike Botts, "OpenGIS® Sensor Model Language (SensorML) Implementation Specification", OGC® 07-000, Open Geospatial Consortium Inc, 2007