

Open Geospatial Consortium Inc.

Date: 2009-07-27

Reference number of this document: OGC 09-082

Version: 0.3.0

Category: OGC® Discussion Paper

Editors: Hsu-Chun James Yu
Zhong-Hung Lee
Cai-Fang Ye
Lan-Kun Chung
Yao-Min Fang

OGC®: Sensor Web Enablement Application for Debris Flow Monitoring System in Taiwan

Copyright © 2009 Open Geospatial Consortium, Inc.

To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>.

Warning

This document is not an OGC Standard. This document is an OGC Discussion Paper and is therefore not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, an OGC Discussion Paper should not be referenced as required or mandatory technology in procurements.

Document type: OpenGIS® Discussion Paper
Document subtype:
Document stage: Approved for public release
Document language: English

This page left intentionally blank.

Contents

	Page
i. Preface	vi
ii. Document term and definitions	vi
iii. Submitting organizations	vi
iv. Document contributor contact points	vi
v. Revision history	vii
vi. Changes to the OGC abstract specification	vii
Forward	viii
Introduction	ix
1. Scope	1
2. Conformance	1
3. Normative references	1
4. Terms and definitions	2
5. Conventions	4
6. Debris Flow Disaster Management Overview	6
6.1 Debris Flow	6
6.2 SWE Introduction	6
7. Architecture of Debris Flow Disaster Management System	8
7.1 DoD Architecture Framework	8
7.2 All View	10
7.3 Operational View	10
7.4 System and Service View	15
7.5 Technical Standards View	17
8. Implementation	18
8.1 Scenario	18
8.2 Architecture Overview	21
8.3 Control: SPS	23
8.4 Notification: WNS	28
8.5 Observation: SOS	32
8.6 Alert: SAS	38
9. Finding and Outlook	43
ANNEX A	44
ANNEX B	45

Figures	Page
Figure 1 - Views of DoD AF.....	9
Figure 2 - OV-1 of debris flow monitoring system.....	11
Figure 3 - OV-2 of debris flow monitoring system.....	12
Figure 4 - OV-5 of debris flow monitoring system.....	13
Figure 5 - OV-7 of debris flow monitoring system.....	13
Figure 6 - SV-1 for debris flow monitoring system.....	15
Figure 7 - SV-2 for debris flow monitoring system.....	16
Figure 8 - SV-4 of debris flow monitoring system	17
Figure 9 - Overview of Scenario	18
Figure 10 - Scenario of SPS	19
Figure 11 - Scenario of SOS	20
Figure 12 - Scenario of SAS	21
Figure 13 - Overview of Architecture	22
Figure 14 - Use case diagram of SPS.....	23
Figure 15 - Sequence diagram of tasking a sensor.....	24
Figure 16 - Use case diagram of WNS.....	28
Figure 17 - Sequence diagram of Register.....	29
Figure 18 - Sequence diagram of DoNotification	31
Figure 19 - Use case diagram of SOS	33
Figure 20 - Sequence diagram of getting observations	34
Figure 21 - Relationships among feature of interest, observed properties, and procedures.....	35
Figure 22 - Procedure of getting big-sized observation data	38
Figure 23 - Use case of SAS	38
Figure 24 - Sequence diagram of Subscription and Notification.....	39

Tables

Page

Table 1 - TV-1 of debris flow monitoring system..... 17

i. Preface

The Open Geospatial Consortium (OGC) is an international industry consortium of more than 380 companies, government agencies, and universities participating in a consensus process to develop publicly available geo-processing specifications.

This profile is contributed by Industrial Technology Research Institute (ITRI), TAIWAN and Geographic Information System Research Center, FCU, TAIWAN. The objective of this document is to demonstrate how to apply OGC Sensor Web Enablement (SWE) specification to Debris Flow Monitoring System.

ii. Document term and definitions

This document uses the specification terms defined in Subclause 5.3 of [OGC 05-008], which is based on the ISO/IEC Directives, Part 2. Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this specification.

iii. Submitting organizations

The following organizations submitted this document to the Open Geospatial Consortium Inc.

Industrial Technology Research Institute

195, Sec. 4, Chung Hsing Rd.,
Chutung, Hsinchu
TAIWAN 31040

Geographic Information Systems Research Center, FCU

100 Wenhwa Rd., Taichung,
TAIWAN

iv. Document contributor contact points

All questions regarding this submission should be directed to the editor. Additional contributors are listed below:

Contributors:

Name	Organization
Kuo-Yu slayer Chuang	ITRI
Zhong-Hung Lee	ITRI
Hsu-Chun James Yu	ITRI
Cai-Fang Ye	ITRI

Lan-Kun Chung	GIS.FCU
Yao-Min Fang	GIS.FCU
Hsiao-Yuan Yin	SWCB

v. Revision history

Date	Release	Editor	Primary clauses modified	Description
2009-06-04	0.1.0	Hsu-Chun James Yu	initial version	Initial version;
2009-07-10	0.3.0	Carl Reed	Various	For publication as DP

vi. Changes to the OGC abstract specification

The OpenGIS® Abstract Specification does not require changes to accommodate the technical contents of this document.

Forward

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

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

Introduction

The OGC Sensor Web Enablement (SWE) suite of standards describe standard interfaces and encodings that provide sensor data exchangeability to web services SWE is also a main component of OGC Web Services (OWS). To implement the full capabilities of the SWE standards, four major services should be provided. The four major services are Sensor Observation Service (SOS), Sensor Planning Service (SPS), Web Notification Service (WNS), and Sensor Alert Service (SAS). All sensor data should be described in XML format. Simple Object Access protocol (SOAP) should be used by each service to provide communicability for each service.

For adopting SWE in the Taiwan Debris Flow Monitoring System, we present how to use XML documents to describe each operation in each service, and how to use XML documents to describe each sensor. The purpose of this document is trying to integrate SWE services, and to provide a tutorial document for SWE developers.

OGC®: Sensor Web Enablement Application for Debris Flow Monitoring System in Taiwan

1. Scope

This application document describes:

- 1) What is a Debris Flow Monitoring System.
- 2) How SWE implements in Debris Flow Monitoring System.
- 3) Tutorial for SWE developers.

2. Conformance

3. Normative references

The following normative documents contain provisions that, through reference in this text, constitute provisions of this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

- [1] OGC 05-008, OGC Web Service Common Specification, version 1.0
- [2] OGC 07-000, OpenGIS® Sensor Model Language (SensorML) Implementation Specification
- [3] OGC 07-022, OpenGIS® Observation & Measurement – Part 1 – Observation Schema
- [4] OGC 07-002r2, OpenGIS® Observation & Measurement – Part 2 – Sampling Features
- [5] OGC 06-009r6, OpenGIS® Sensor Observation Service Implementation Specification
- [6] OGC 07-014r3, OpenGIS® Sensor Planning Service Implementation Specification
- [7] OGC 06-028r5, OpenGIS® Sensor Alert Service Implementation Specification
- [8] OGC 06-095r1, OpenGIS® Web Notification Service Implementation Specification
- [9] OGC 03-109r1 OGC Web Map Service Interface

4. Terms and definitions

For the purposes of this specification, the definitions specified in Clause 4 of the OWS Common Implementation Specification [OGC 05-008] shall apply. In addition, the following terms and definitions apply.

Debris Flow

A debris flow is a fast moving mass of unconsolidated, saturated debris that looks like flowing concrete. They differentiate from a mudflow by terms of the viscosity of the flow. Flows can carry clasts ranging in size from clay particles to boulders, and also often contains a large amount of woody debris. Flows can be triggered by large amounts of rainfall, or glacial melt, or a combination of the two. Speed of debris flows can vary from 1mph to 35mph in extreme conditions. Variables in the conditions that effect the debris flow can include slope, and available loose sediment. Debris flow are extremely destructive to life and property.

Architecture (of a system) (ISO/IEC 10746-2:1996)

Set of rules to define the structure of a system and the interrelationships between its parts

Application (derived from <http://www.opengeospatial.org/resources/?page=glossary>)

Use of capabilities, including hardware, software and data, provided by an information system specific to the satisfaction of a set of user requirements in a given application domain.

End user (OGC 07-097)

Members of agencies (e.g. civil or environmental protection agencies) or private companies that are involved in an application domain (e.g. risk management) and that use the applications built by the system users.

Event

Any happening of interest that can be observed from within a computer.

Implementation (<http://www.opengeospatial.org/resources/?page=glossary>)

Software package that conforms to a standard or specification. A specific instance of a more generally defined system.

Notification

Reifies an event as a datum, i.e. it contains data describing the event .

Observation (OGC 07-022)

Act of observing a property or phenomenon, with the goal of producing an estimate of the value of the property.

Operation (ISO 19119)

Specification of a transformation or query that an object may be called to execute.

Parameter

Variable whose name and value are included in an operation **request** or **response**

Representation

Comprises any useful information about the current state of a resource.

Sensor

Entity that provides information about an observed property at its output. A sensor uses a combination of physical, chemical or biological means in order to estimate the underlying observed property. At the end of the measuring chain electronic devices produce signals to be processed.

Service (ISO 19119:2005)

Distinct part of the functionality that is provided by an entity through interfaces.

System User (OGC 07-097)

Provider of services that are used for an application domain as well as IT architects, system developers, integrators and administrators that conceive, develop, deploy and run applications for an application domain.

User (OGC 07-097)

Human acting in the role of a system user or end user.

Viewpoint (RM-ODP)

Subdivision of the specification of a complete system, established to bring together those particular pieces of information relevant to some particular area of concern during the design of the system.

Web Service

Self-contained, self-describing, modular service that can be published, located, and invoked across the Web. A Web service performs functions, which can be anything from simple requests to complicated business processes. Once a Web service is deployed, other applications (and other Web services) can discover and invoke the deployed service.

5. Conventions

Abbreviated terms

Some frequently used abbreviated terms:

CCD	Charge-Coupled Device
DoD AF	Department of Defense Architecture Framework
HTTP	HyperText Transport Protocol
ISO	International Organisation for Standardisation
LBS	Location Base Service
OGC	Open Geospatial Consortium
O&M	Observation and Measurement
SAS	Sensor Alert Service
SensorML	Sensor Model Language
SOAP	Simple Object Access Protocol
SOS	Sensor Observation Service
SPS	Sensor Planning Service
SWCB	Soil and Water Conservation Bureau
SWE	Sensor Web Enablement
UUID	Universal Unique Identifier
WFS	Web Feature Service
WMS	Web Map Service
WNS	Web Notification Service
XML	eXtensible Markup Language

Document terms and definitions

The following specification terms and definitions are used in this document:

- a) shall – verb form used to indicate a requirement to be strictly followed to conform to this specification, from which no deviation is permitted
- b) should – verb form used to indicate desirable ability or use, without mentioning or excluding other possibilities
- c) may – verb form used to indicate an action permissible within the limits of this specification
- d) can – verb form used for statements of possibility
- e) informative – a part of a document that is provided for explanation, but is not required
- f) normative – a part of a standards document that is required
- g) annex – an auxiliary part of a document
- h) Open GeoSMS – an exchangeable short message format for LBS application.

6. Debris Flow Disaster Management Overview

6.1 Debris Flow

Taiwan is located at the collision boundary of the Philippine ocean plate and the Eurasian plate. The mountain terrain is precipitous and the region, on the whole, is characterized by fragile rocks and frequent seismic activity. In addition, the concentrated torrential rainfall brought by typhoons and Mei-Yu cause extensive disasters. There are two reasons for the occurrence of debris flow after a strong earthquake. One is that the land collapses after earthquake and the soil gets mixed with groundwater or surface runoff. The second reason is that many crevices are formed in the earth surface after earthquake and hence, when the groundwater level increases or surface runoff concentrates, the land collapses and debris flow occurs. After the 921 earthquake in 1999, six severe typhoons namely, Xangsane Typhoon (2000), Toraji, Nari and Lekima Typhoons (2001), and Mindulle and Aere Typhoons (2004), caused serious disasters in the central Taiwan. Since 2002, the Soil and Water Conservation Bureau, which is responsible for the conservation and administrative management of hillside in Taiwan, has been cooperated with Feng Chia University. Together, they have successively carried out the establishment and maintenance of 13 fixed debris flow monitoring stations over the island and 2 mobile debris flow monitoring stations.

The advanced monitoring instruments include rain gauge, wire sensor, geophone, and CCD camera. Rain gauge is used to record on-site rainfall. At the moment, the warning model for the debris flow alert uses rainfall intensity and accumulated precipitation as warning indexes to determine whether rainfall has reached the threshold and thereby release timely red and yellow alerts to high risk areas where debris flows are likely to occur. As debris flows gushing down the channel, the flow will break the wire sensor placed in the spillway of sabo dam, hence indicating the occurrence of debris flow. Besides, when debris flows occur, the geophone can detect the ground vibration generated by the collision between boulders and channel bed. The result of wavelet transform analysis can then serve as references to determine the occurrence of debris flow. Finally through the CCD camera, the hydrological process of debris flow can be vividly recorded.

6.2 SWE Introduction

Sensor Web Enablement standards that have been built and prototyped by members of the OGC include the following pending OpenGIS Specifications:

1. Observations & Measurements (O&M) : The general models and XML encodings for observations and measurements made using sensors.

2. **Sensor Model Language (SensorML)** : standard models and XML Schema for describing the processes within sensor and observation processing systems; provides information needed for discovery, georeferencing, and processing of observations, as well as tasking sensors and simulations.
3. **Sensor Observation Service (SOS)** : An open interface for a service by which a client can obtain observations and sensor and platform descriptions from one or more sensors.
4. **Sensor Planning Service (SPS)** : An open interface for a service by which a client can determine the feasibility of collecting data from one or more sensors processes.
5. **Sensor Alert Service (SAS)** : An open interface for a web service for publishing of and subscribing to deliverable alerts from sensor or simulation systems.
6. **Web Notification Service (WNS)** : An open interface for a service by which a client may conduct asynchronous dialogues (message interchanges) with one or more other services.

7. Architecture of Debris Flow Disaster Management System

7.1 DoD Architecture Framework

Architectures within the Department of Defense (DoD) are created for a number of reasons. From a compliance perspective, the DoD's development of architectures is compelled by law and policy (i.e., Clinger-Cohen Act, Office of Management and Budget (OMB) Circular A-130). From a practical perspective, experience has demonstrated that the management of large organizations employing sophisticated systems and technologies in pursuit of joint missions demands a structured, repeatable method for evaluating investments and investment alternatives, as well as the ability to effectively implement organizational change, create new systems, and deploy new technologies. Towards this end, the DoD Architecture Framework (DoDAF) was established as a guide for the development of architectures. (For the purposes of this document the terms DoDAF and framework are synonymous.)

The DoDAF provides the guidance and rules for developing, representing, and understanding architectures based on a common denominator across DoD, Joint, and multinational boundaries. It provides insight for external stakeholders into how the DoD develops architectures. The DoDAF is intended to ensure that architecture descriptions can be compared and related across programs, mission areas, and, ultimately, the enterprise, thus, establishing the foundation for analyses that supports decision-making processes throughout the DoD.

As the Department takes appropriate strides to ensure advancement of the Information Technology (IT) environment, it is essential for the DoDAF to transform to sufficiently support new technologies. A significant evolution occurring today is the Department's transformation to a new type of information intensive warfare known as Net-Centric Warfare (NCW). NCW focuses on generating combat power from the effective linking or networking of the warfighting enterprise, and making essential information available to authenticated, authorized users when and where they need it.

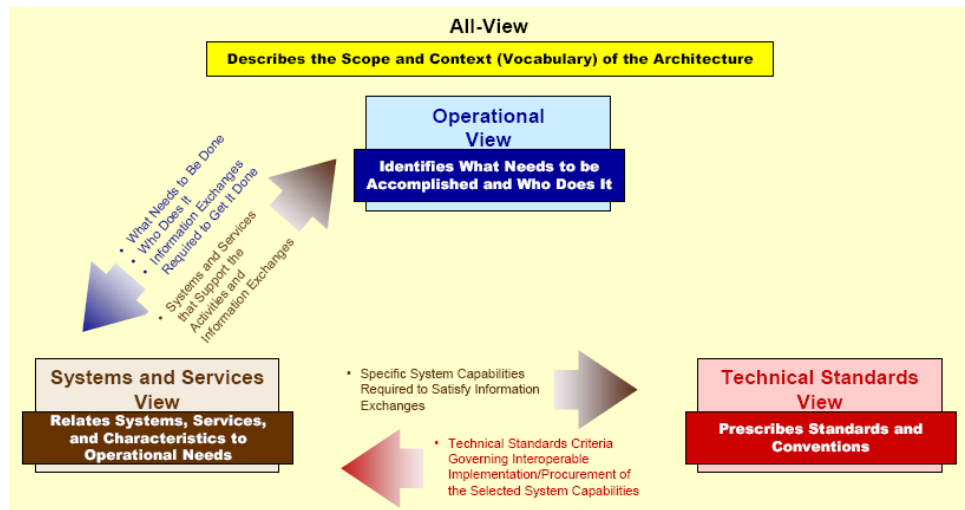


Figure 1 - Views of DoD AF

Figure 1 represents the information that links the operational view, systems and services view, and technical standards view. The three views and their interrelationships driven – by common architecture data elements – provide the basis for deriving measures such as interoperability or performance, and for measuring the impact of the values of these metrics on operational mission and task effectiveness.

- **Definition of the Operational View**

The OV captures the operational nodes, the tasks or activities performed, and the information that must be exchanged to accomplish DoD missions. It conveys the types of information exchanged, the frequency of exchange, which tasks and activities are supported by the information exchanges, and the nature of information exchanges.

- **Definition of the Systems and Services View**

The SV captures system, service, and interconnection functionality providing for, or supporting, operational activities. DoD processes include warfighting, business, intelligence, and infrastructure functions. The SV system functions and services resources and components may be linked to the architecture artifacts in the OV. These system functions and service resources support the operational activities and facilitate the exchange of information among operational nodes.

- **Definition of the Technical Standards View**

The TV is the minimal set of rules governing the arrangement, interaction, and interdependence of system parts or elements. Its purpose is to ensure that a system satisfies a specified set of operational requirements. The TV provides the technical systems implementation guidelines upon which engineering specifications are based, common building blocks are established, and product lines are developed. It includes a collection of the technical standards, implementation conventions, standards options, rules, and criteria that can be organized into profile(s) that govern systems and system or service elements for a given architecture.

Definition of the All-View

There are some overarching aspects of an architecture that relate to all three views. These overarching aspects are captured in the AV products. The AV products provide information pertinent to the entire architecture but do not represent a distinct view of the architecture. AV products set the scope and context of the architecture. The scope includes the subject area and time frame for the architecture. The setting in which the architecture exists comprises the interrelated conditions that compose the context for the architecture. These conditions include doctrine; tactics, techniques, and procedures; relevant goals and vision statements; concepts of operations (CONOPS); scenarios; and environmental conditions.

7.2 All View

All view is the description of scope, purpose, intended users, environment depicted, analytical findings.

Architecture Project Identification

Debris flow monitoring system has been developing for 9 years since 2000. Soil and water conservation bureau, the competent authority for debris flow in Taiwan, has been cooperating with GIS Center, Feng Chia University to develop the whole architecture since the era without available open standard. Prof Lee, Bean-Jean and Prof. Chou, Tien-Yin are the main architects of this project. Interoperability, the nature of monitoring system and disaster prevention task, should be emphasized and adopted into the existing architecture, so the main direction of the new debris flow monitoring system is to step toward to an open standard based architecture in order to facing the highly interoperable world.

Purpose and viewpoint

Debris flow monitoring system is established to meet the requirement of disaster management. By monitoring, collecting and analyzing sensor data, decision maker can make a rational decision of debris flow alert/evacuation command for those residents who live in the impact zone of debris flow river, thus, this architecture is developed by the viewpoint of decision maker.

7.3 Operational View

OV-1

The main concept of Debris Flow Monitoring System is to connect various sensors(resource satellite, fixed monitoring station, mobile monitoring station, UAV, fixed-wing airplane, digital camera, etc.) in order to establish a network of ubiquitous monitoring to command, control, communicate, survey, reconnoiter by computer to give intelligence to decision maker. Figure 2 shows the general concept of how Debris Flow

Monitoring System works in OV-1.

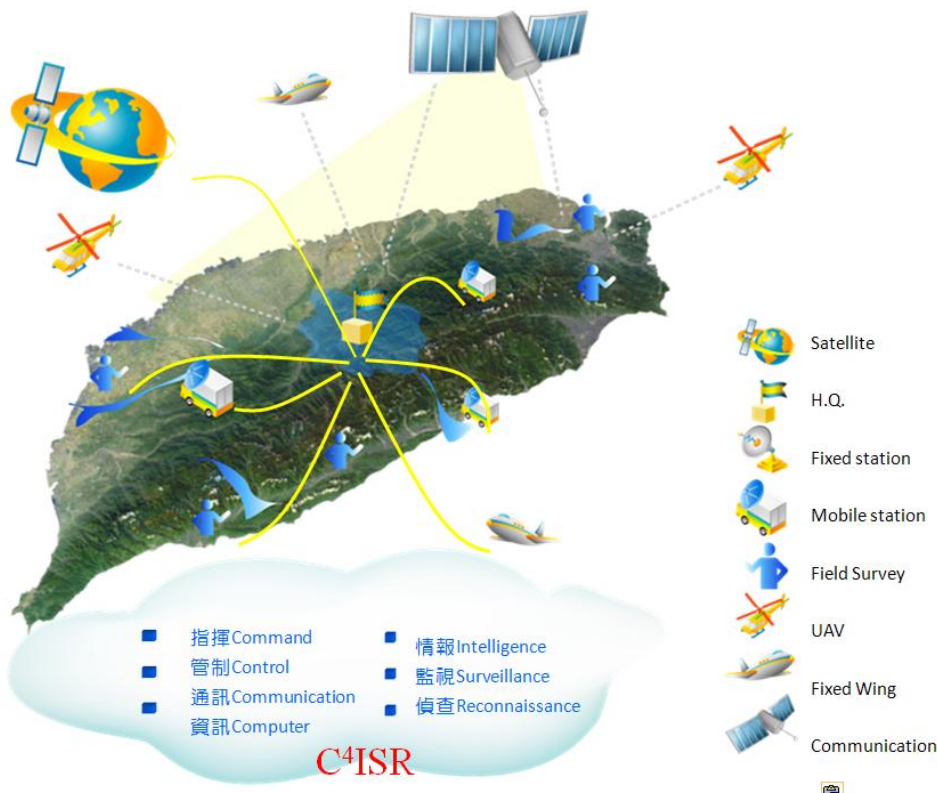


Figure 2 - OV-1 of debris flow monitoring system

OV-2

The OV-2 graphically depicts the operational nodes (or organizations) with needlines between those nodes that indicate a need to exchange information. The graphic includes internal operational nodes (internal to the architecture) as well as external nodes. OV-2 is intended to track the need to exchange information from specific operational nodes (that play a key role in the architecture) to others. OV-2 does not depict the connectivity between the nodes. Figure 3 illustrates several critical operational nodes in debris flow monitoring system.

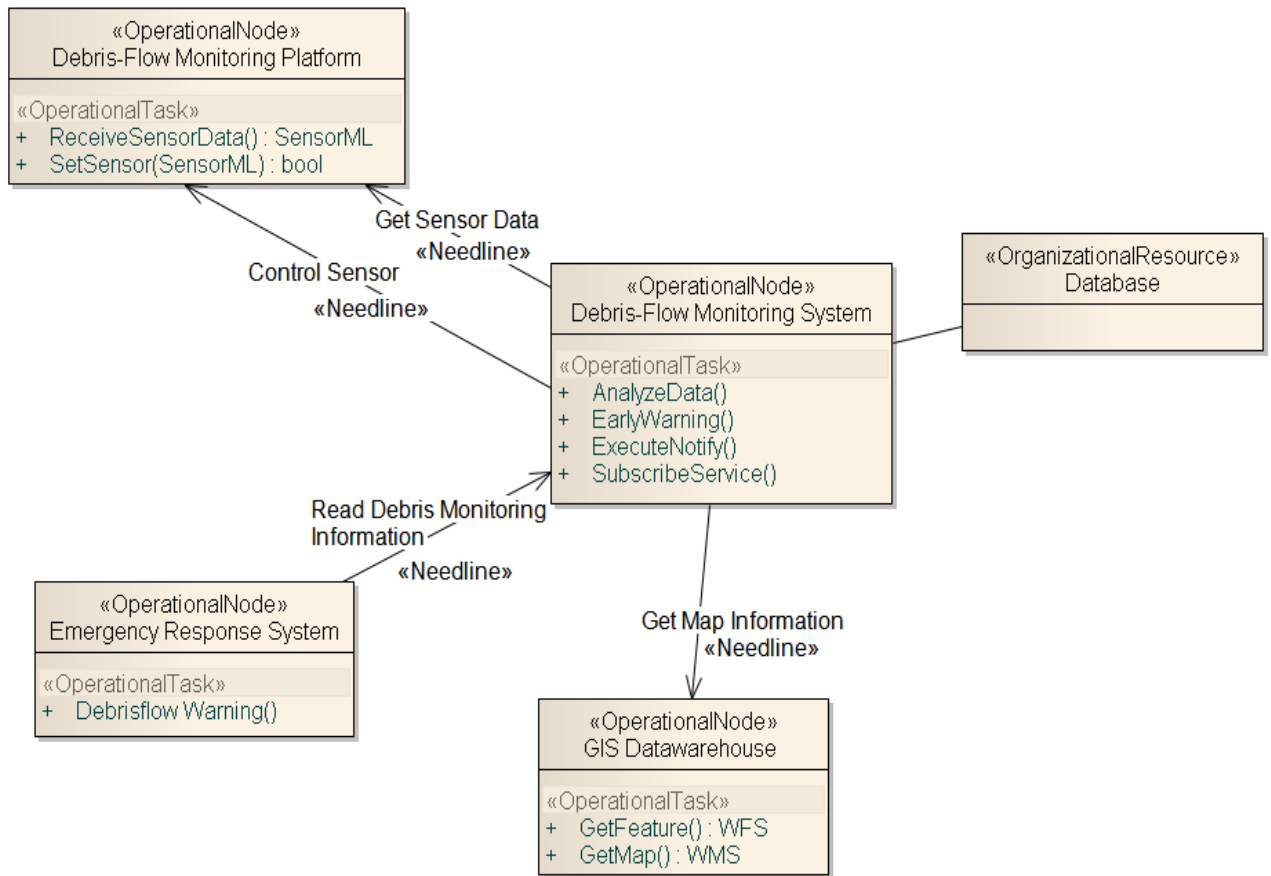


Figure 3 - OV-2 of debris flow monitoring system

OV-5

The OV-5 describes the operations that are normally conducted in the course of achieving a mission or a business capability. It describes capabilities, operational activities (or tasks), input and output (I/O) flows between activities, and I/O flows to/from activities that are outside the scope of the architecture. Figure 4 illustrates a classical relationship between upstream sensors and downstream decision support activities of debris flow monitoring system

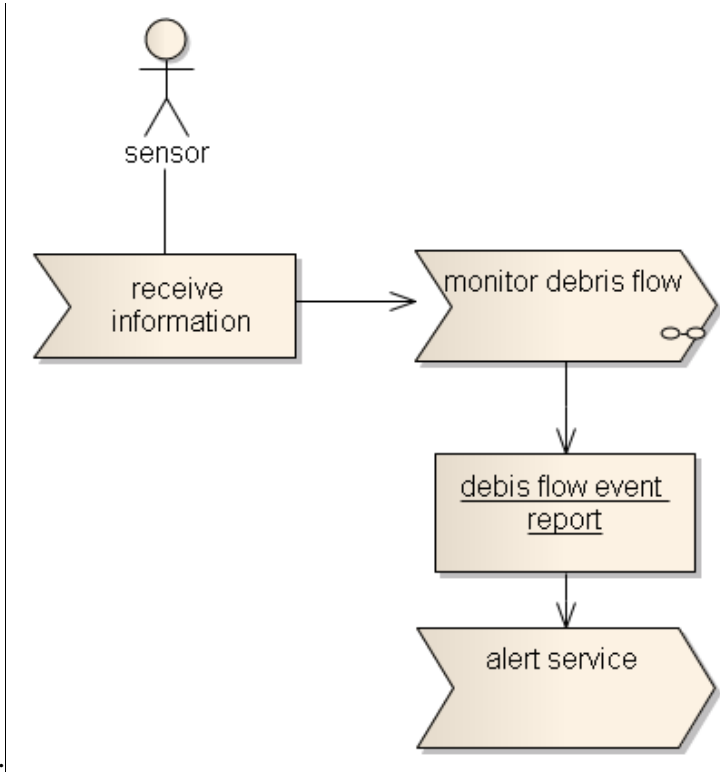


Figure 4 - OV-5 of debris flow monitoring system

OV-7

The OV-7 describes the structure of an architecture domain’s system data types and the structural business process rules (defined in the architecture’s OV) that govern the system data. It provides a definition of architecture domain data types, their attributes or characteristics, and their interrelationships. Figure 5 shows four main sensors types of a classical debris flow monitoring station.

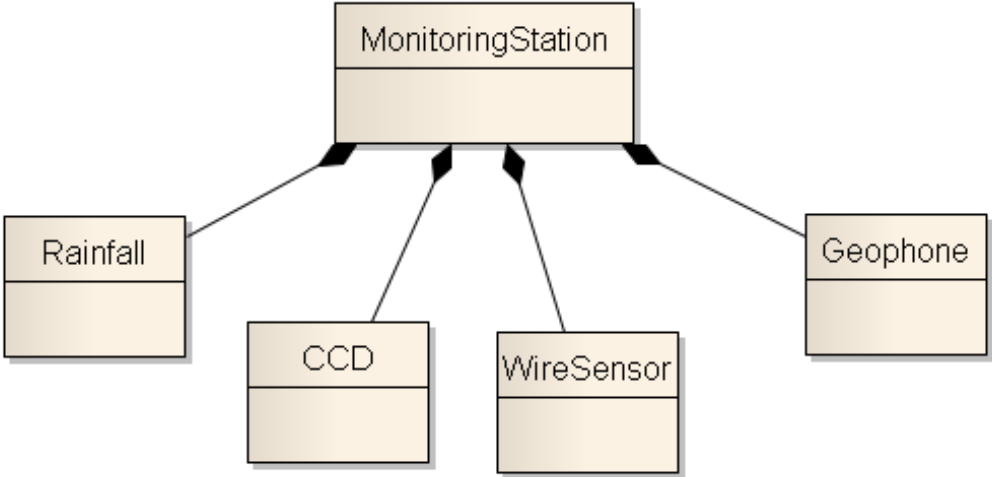


Figure 5 - OV-7 of debris flow monitoring system

7.4 System and Service View

SV-1

The SV-1 depicts systems nodes and the systems resident at these nodes to support organizations/human roles represented by operational nodes of the OV-2. SV-1 also identifies the interfaces between systems and systems nodes. Figure 6 illustrates three sub-systems in a debris flow monitoring system, debris flow information platform which supports standard interfaces to emergency response system to multiple decision making usage, spatial data warehouse which publishes GIS maps to emergency response system to assist some spatial analysis tasks.

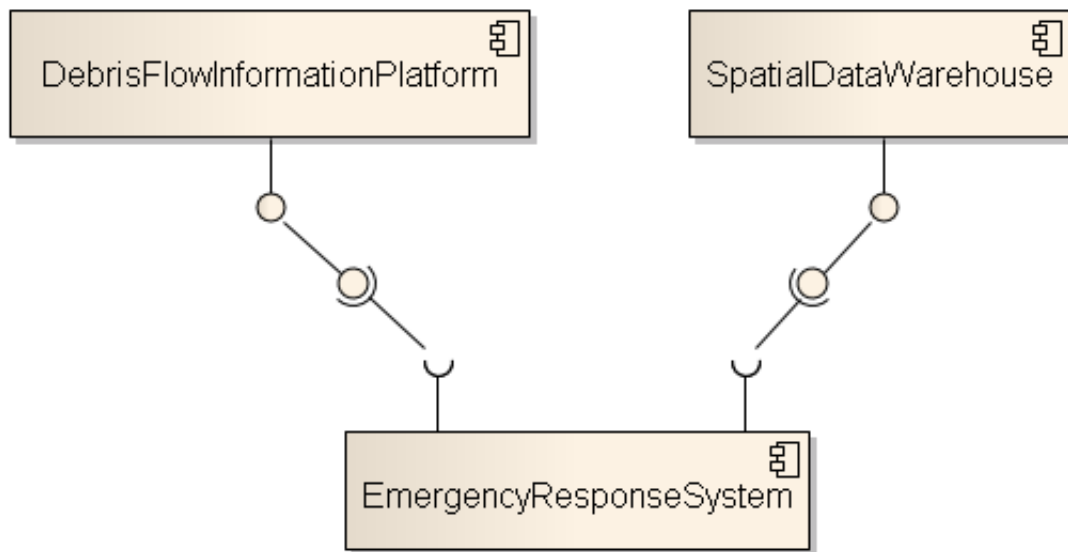


Figure 6 - SV-1 for debris flow monitoring system

SV-2

The SV-2 depicts pertinent information about communications systems, communications links, and communications networks. SV-2 documents the kinds of communications media that support the systems and implements their interfaces as described in SV-1. Thus, SV-2 shows the communications details of SV-1 interfaces that automate aspects of the needlines represented in OV-2. Figure 7 is one of system and service view of debris flow monitoring system.

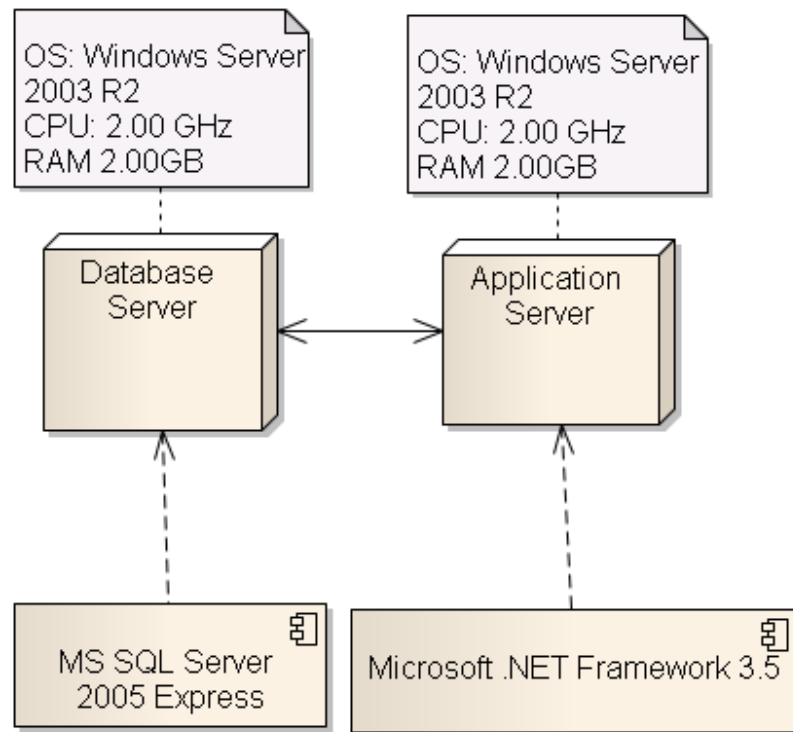


Figure 7 - SV-2 for debris flow monitoring system

SV-4

The SV-4a documents system functional hierarchies and system functions, and the system data flows between them. The SV-4 from DoDAF v1.0 is designated as 'SV-4a' in DoDAF v1.5. Although there is a correlation between OV-5 or business-process hierarchies and the system functional hierarchy of SV-4a, it needs not be a one-to-one mapping, hence, the need for the Operational Activity to Systems Function Traceability Matrix (SV-5a), which provides that mapping. Figure 8 shows the relationship between system functions and system data flow in a UML use case diagram.

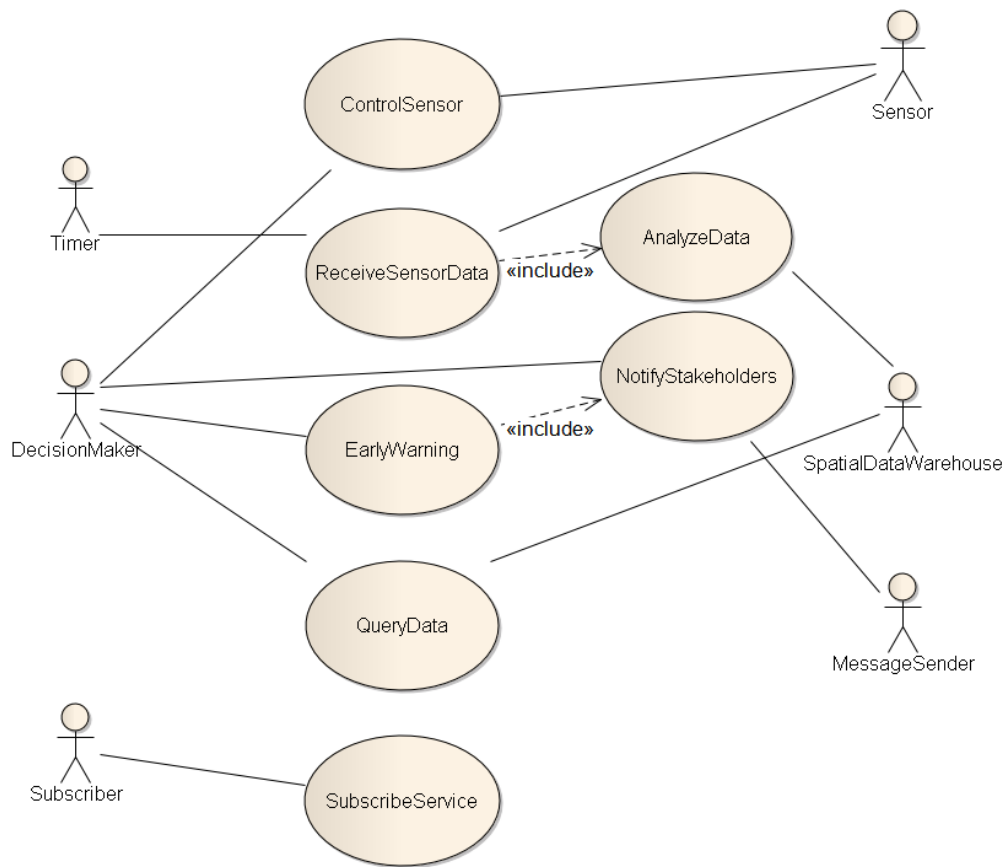


Figure 8 - SV-4 of debris flow monitoring system

7.5 Technical Standards View

The TV-1 (Table 1) collects the various systems standards rules that implement and sometimes constrain the choices that can be made in the design and implementation of an architecture. TV-1 serves as the bridge between the SV and TV.

Table 1 - TV-1 of debris flow monitoring system

Standard Name	Version	Organization
Sensor Model Language Implementation Specification	1.0.0	OGC
Observations and Measurements	1.0	OGC
Sensor Observation Service	0.1.4	OGC
Sensor Planning Service Implementation Specification	1.0	OGC
Sensor Alert Service Implementation Specification	0.9.0	OGC
Web Notification Service	0.1.0	OGC
Web Map Service	1.3.0	OGC
Web Feature Service	1.1	OGC
OGC KML	2.2.0	OGC
WS-BPEL	2.0	OASIS
WS-Trust	1.3	OASIS
WS-Security	1.0	OASIS

8 Implementation

8.1 Scenario

From the scenario in Figure 9, in order to provide prediction and alarm information about Debris Flow, we use sensors to observe the Chiu-fen-er-shan environment. Having the observation data, the user can get the alert before the Debris Flow happen, or get the observation data he interested in.

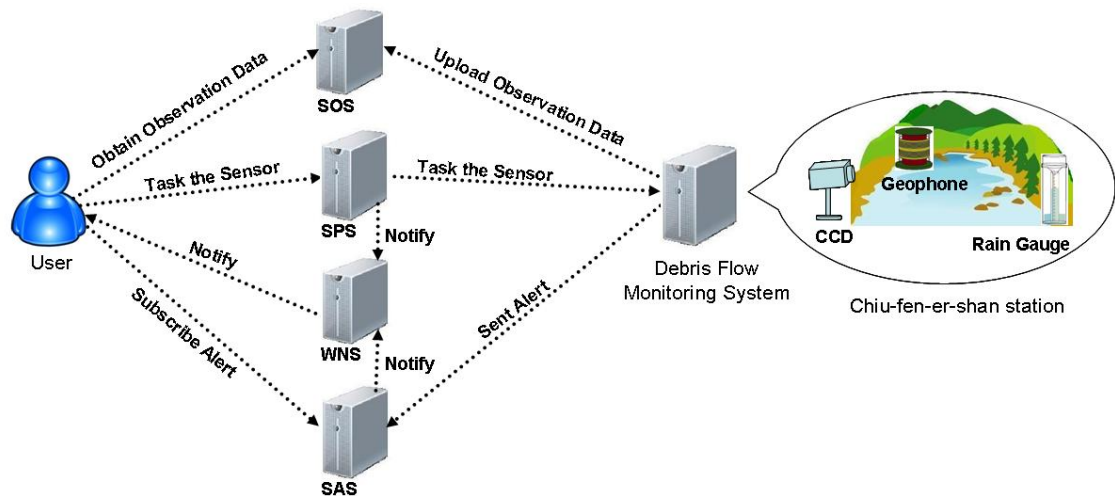


Figure 9 - Overview of Scenario

To complete the Debris Flow monitoring, the user should know

1. How to control the sensors to observe the station environment?
2. How to obtain the observation data?
3. How could the user get the notification?

In SWE, it provides several services to fulfill the monitoring. These services include SPS, SOS, SAS, WNS...etc.

The next paragraph illustrates how to adopt SWE to Debris Flow Monitoring System.

1. We use Rain Gauge, CCD and Geophone to observe the Chiu-fen-er-shan station environment. Debris Flow Monitoring System is response for managing these sensors. For each service, Debris Flow Monitoring System plays different roles as follows.
 - i. For SPS, it is taking charge of controlling. According to the task that is submitted by SPS, Debris Flow Monitoring System will control the relevant sensors to do observing.

- ii. For SOS, it is taking charge of collecting. Debris Flow Monitoring System collects observation data from sensors and then uploads to SOS.
 - iii. For SAS, it is taking charge of decision. Debris Flow Monitoring System has domain knowledge to decide whether debris flow will happen. If the answer is “Yes”, it will send an alert to SAS.
2. The user uses SPS to task these sensors. In order to avoid that the user arbitrarily changes the parameters of sensors, this functionality is mainly used by Administrator.
 3. There are two ways the user can obtain the observation data
 - Using SAS to subscribe the alert that he concerns about. And then, the user will get the alert sent by WNS when the event happens.
 - Using SOS to get the observation data.

SPS scenario

Figure 10 shows the scenario of SPS. The major work of SPS is tasking the sensors.

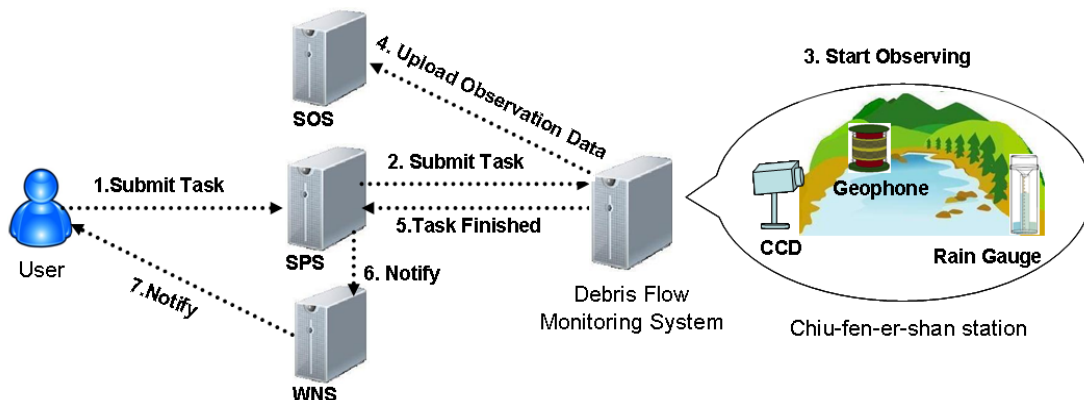


Figure 10 - Scenario of SPS

Before tasking the sensors, the user needs to know

- What information of sensors and phenomenon are provided by SPS?
- How to task the sensors?
- How and Where the observation data can be accessed?

The user can get all the answers through SPS operations that we will introduce later.

The main procedure of tasking a sensor as follows.

1. The user sets the task parameters of the sensor, then submits the task to SPS.

2. SPS transfers the task to Chiu-fen-er-shan station.
3. The sensor which was tasked starts observing.
4. Chiu-fen-er-shan station collects these observation data and uploads to SOS.
5. Chiu-fen-er-shan station sends notification to SPS when the task is finished.
6. SPS notifies WNS to send notification to the user.
7. The user receives the notification sent by WNS.

SOS scenario

Figure 11 shows the scenario of SOS. The major work of SOS is to obtain observation data.

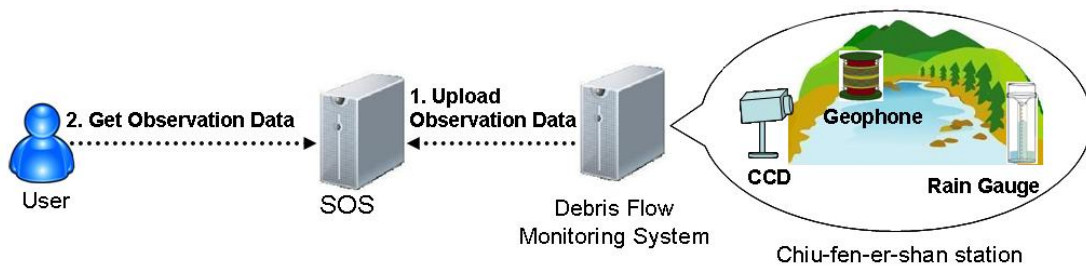


Figure 11 - Scenario of SOS

Before getting observation data, the user needs to know

- What observation data are provided?
- Which sensor provides the observation data?
- What abilities the sensor has?
- What filters the user can set to get observation data?

The user can get all the answers through SOS operations that we will introduce later. With the above-mentioned information, the user can get observation data that the user wants to retrieve from SOS.

SAS scenario

Figure 12 shows the scenario of SAS. The major work of SAS is to process the alert subscriptions.

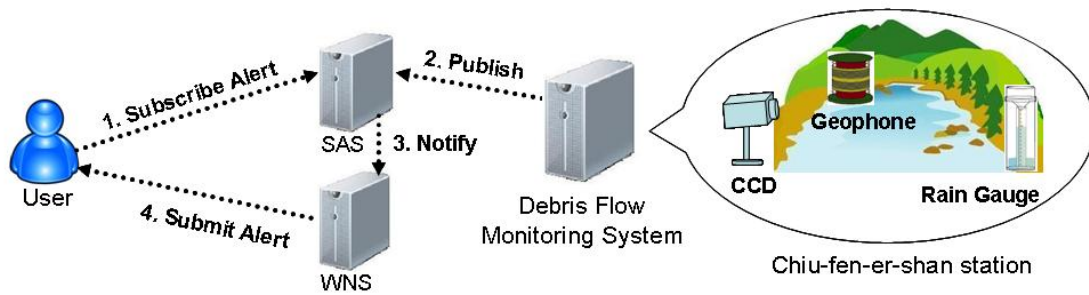


Figure 12 - Scenario of SAS

Before subscribing alert, the user needs to know how many and what kinds of alert events can be subscribed through SAS operations. With the above-mentioned information, the main procedure of alert notification as follows.

1. The user submits the subscription.
2. Chiu-fen-er-shan station publishes notification while alert was invoked.
3. SAS notifies WNS to send notification to the user.
4. The user receives the alert sent by WNS.

8.2 Architecture Overview

From previous section, we know the web services in difference scenarios have different purpose, and provide different functions. We can follow the scenario to design SWE architecture for Debris Flow Monitoring System. Figure 13 demonstrates how we design web services and which operations will be used in each web service. The architecture graphic shows below:

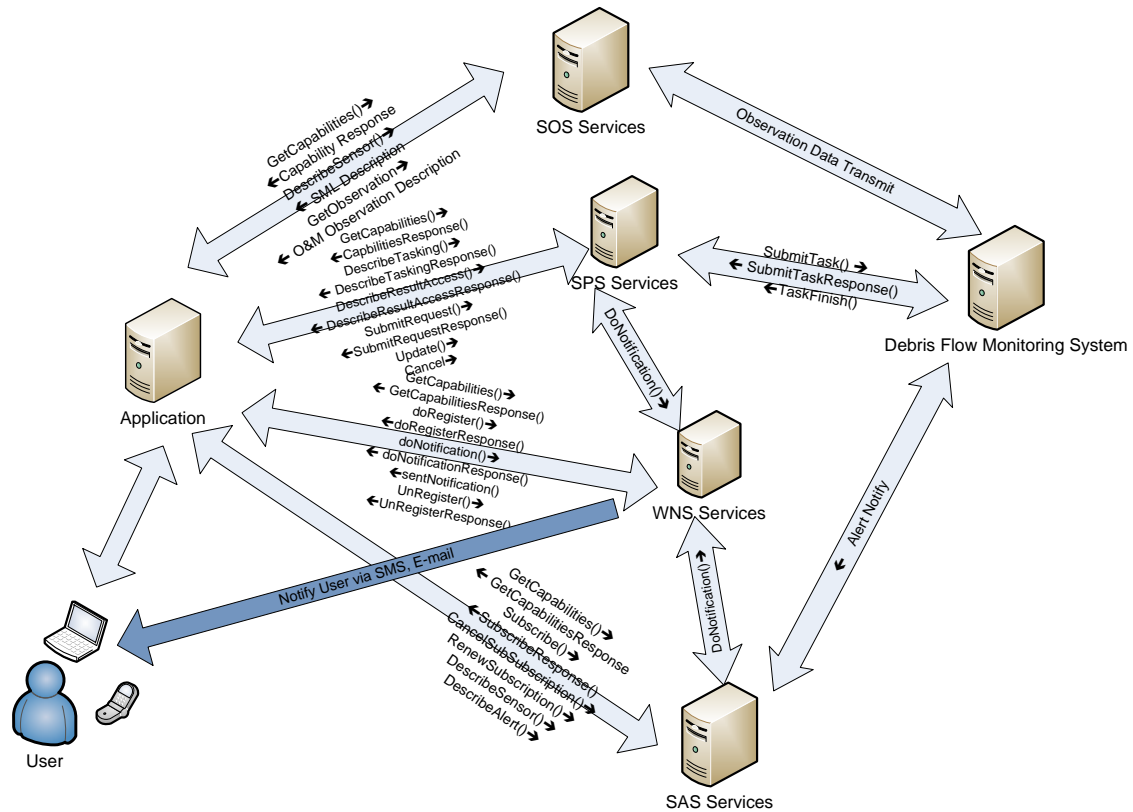


Figure 13 - Overview of Architecture

We follow section 8.1 scenarios to design SWE architecture for debris flow monitoring System, and for each web service, SOS, SPS, WNS, and SAS, operations describe as follows:

1. SOS: major work is to obtain sensors data, and SOS has 3 operations a) GetCapabilities, b) DescribeSensor c) GetObservation.
2. SPS: major work is to task sensor, and SPS has 6 operations a) GetCapabilities b) DescribeTasking c) DescribeResultAccess d) SubmitRequest e) Update f) Cancel
3. WNS: major work is to notice the user, and WNS has 4 operations a) GetCapabilities b) DoRegister c) DoNotification d) UnRegister
4. SAS: major work is to provide alert notice from the user subscribe, and SAS has 6 operations a) GetCapabilities b) Subscribe c) CancelSubscription d) RenewSubscription e) DescribeSensor f) DescribeAlert

Each operation will be described in following sections.

8.3 Control: SPS

Figure 14 is the Use Case Diagram of SPS. This diagram shows that SPS mainly provides four functions for the user. They are *Obtain Service Capabilities*, *Obtain Result Access Information*, *Obtain Task Information* and *Submit Task*. In *Obtain Result Access Information* use case, the user can retrieve information about where observation data will be accessed. Figure 15 illustrates how the user interacts with SPS.

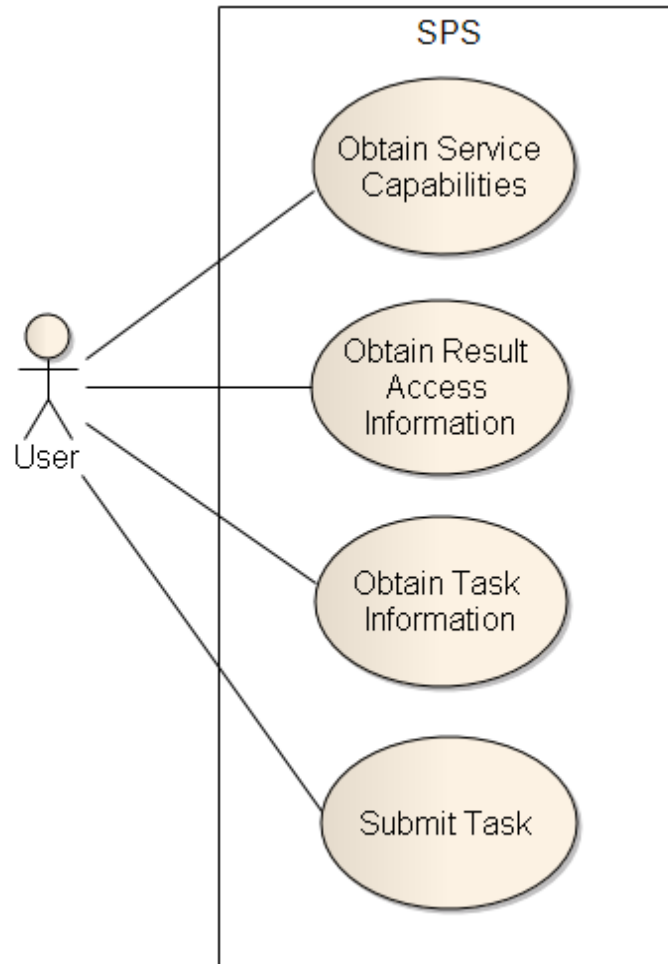


Figure 14 - Use case diagram of SPS

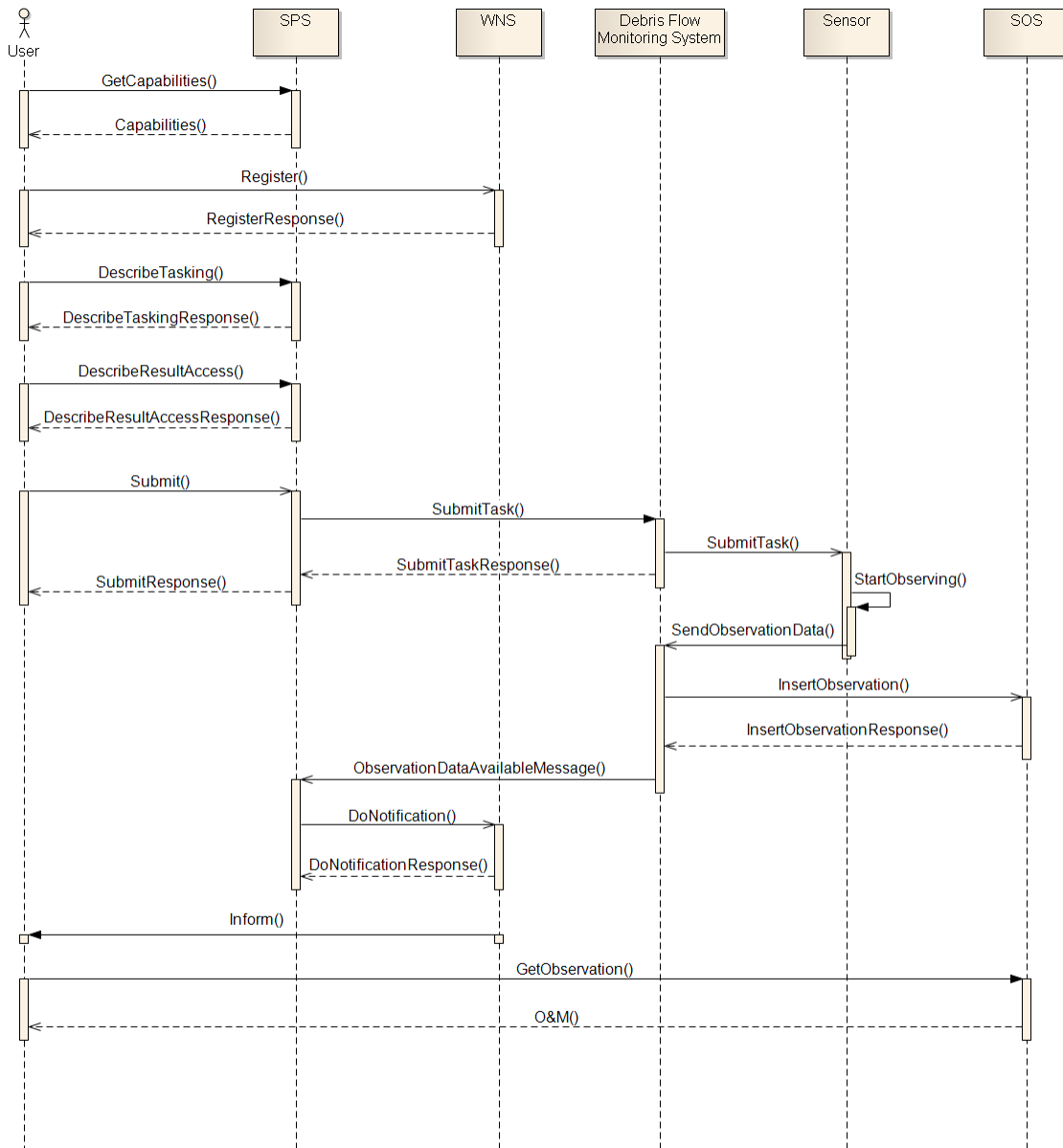


Figure 15 - Sequence diagram of tasking a sensor

Step 1: Obtain SPS service capabilities (GetCapabilities)

Using SPS to task a sensor, first, the user needs to know what capabilities are provided by SPS. The main work of SPS is tasking sensors, so which sensor or phenomenon can be tasked is the major information the user wants to know from *Capabilities* response. In SPS *Capabilities* response, there are two ways help the user to find out the capabilities of SPS. First, through *SensorOfferingList*, the user can know SPS provides which sensors, and each sensor’s information. Vice versa, the user can know which phenomenon observation is provided by which sensor through *PhenomenonOfferingList*. For example,

<p>Capabilities <Contents></p>
--

```

<SensorOfferingList>
  <SensorOffering>
    <AreaOfService>
      <gml:pos>23.950032 120.854216</gml:pos>
    </AreaOfService>
    <Phenomenon>urn:ogc:def:phenomenon:OGC_TW:Rainfall</Phenomenon>
    <SensorDefinition>http://ogc.org.tw/sml/rainSML.xml</SensorDefinition>
    <SensorID>urn:ogc:object:Sensor:OGC_TW:bc14680e-c9ad-484b-ae5f-
      0c97bda5862d</SensorID>
    </SensorOffering>
  </SensorOfferingList>
  <PhenomenonOfferingList>
    <PhenomenonOffering>
      <Phenomenon>urn:ogc:def:phenomenon:OGC_TW:Rainfall</Phenomenon>
      <SensorID>urn:ogc:object:Sensor:OGC_TW:bc14680e-c9ad-484b-ae5f-
        0c97bda5862d</SensorID>
    </PhenomenonOffering>
  </PhenomenonOfferingList>
</Contents>

```

If the user finds the information could fulfill his needs, then he should do **Register** in WNS before submitting the task to get the notification of the task information. The relevant information of WNS will be specified in 8.4.

Step 2: Obtain Task Information (DescribeTasking)

After knowing which sensors SPS provides, the user can obtain the sensor tasking abilities through **DescribeTasking** request. Considering that the user may interested in more than one sensor in one time, in **DescribeTasking** request, the user can list one to many *sensorIDs* to get all the task description back. In **DescribeTaskingResponse** response, each sensor has its **taskingDescriptor** where the user can find out information of task parameters of the sensor. For example,

DescribeTasking

```

<?xml version="1.0" encoding="UTF-8"?>
<DescribeTasking service="SPS" version="1.0.0"
  xmlns="http://www.opengis.net/sps/1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/sps/1.0 http://schemas.opengis.net/sps/1.0.0/spsAll.xsd">
  <sensorID>urn:ogc:object:Sensor:OGC_TW:bc14680e-c9ad-484b-ae5f-0c97bda5862d</sensorID>
  <sensorID>urn:ogc:object:Sensor:OGC_TW:8490a2e3-abaf-4a4e-bae2-6875a61a2a5a</sensorID>
  <sensorID>urn:ogc:object:Sensor:OGC_TW:afe271b3-1cbb-40c9-8d95-4e1dac09a373</sensorID>
</DescribeTasking>

```

DescribeTaskingResponse

```

<taskingDescriptor>
  <sensorID>urn:ogc:object:Sensor:OGC_TW:bc14680e-c9ad-484b-ae5f-
    0c97bda5862d</sensorID>
  <InputDescriptor parameterID="rain-sampling-frequency" updateable="false" use="required">
    <gml:description>How often rain gaugu send date back</gml:description>
    <definition>
      <commonData>
        <swe:Quantity definition="urn:ogc:def:parameter:OGC_TW:sampling-
          frequency">

```

```

        <swe:uom code="Hz"/>
      </swe:Quantity>
    </commonData>
  </definition>
</InputDescriptor>
<InputDescriptor parameterID="threshold" updateable="true" use="required">
  <gml:description>threshold to trigger Geophone</gml:description>
  <definition>
    <commonData>
      <swe:Quantity definition="urn:ogc:def:parameter:OGC_TW:threshold">
        <swe:uom code="mm"/>
      </swe:Quantity>
    </commonData>
  </definition>
</InputDescriptor>
</taskingDescriptor>

```

Step 3: Obtain Result Access Information (DescribeResultAccess)

Before submitting the task of the sensor, the user needs to know whether the observation data can be accessible through *DescribeResultAccess* operation. In *DescribeResultAccess* request, the user provides either *sensorID* or *taskID* to query whether the observation data are accessible. If yes, in *DescribeResultAccessRequestResponse* response will give information of the service which provides the observation data; if no, it will give reasons to explain why the observation data are not available. Notice that in *DescribeResultAccess* request, the user can only request for one *sensorID* or one *taskID*, because the observation data of each *sensorID* or *taskID* may be provided by different services. From the following example we can know the observation data of the *sensorID* can be accessible by SOS.

DescribeResultAccess

```

<?xml version="1.0" encoding="UTF-8"?>
<DescribeResultAccess service="SPS" version="1.0.0"
xmlns="http://www.opengis.net/sps/1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/sps/1.0 http://schemas.opengis.net/sps/1.0.0/spsAll.xsd">
  <sensorID>urn:ogc:object:Sensor:OGC_TW:8490a2e3-abaf-4a4e-bae2-6875a61a2a5a</sensorID>
</DescribeResultAccess>

```

DescribeResultAccessRequestResponse

```

<?xml version="1.0" encoding="UTF-8"?>
<DescribeResultAccessRequestResponse
xmlns="http://www.opengis.net/sps/1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/sps/1.0 http://schemas.opengis.net/sps/1.0.0/spsAll.xsd">
  <service>
    <ServiceType>SOS</ServiceType>
    <ServiceURL>http://ogc.org.tw/sos</ServiceURL>
  </service>
</DescribeResultAccessRequestResponse>

```

Step 4: Submit the Assignment (Submit)

With all the information above, the user can submit the task to Debris Flow Monitoring System to task the sensor in two ways. First is providing *feasibilityID*. If the user decides to use *feasibilityID*, he needs to send *GetFeasibility* request first. Second is providing *sensorID* with all *InputParameters* as our example.

In *Submit* request, the user can set the *notificationID* of users who want to receive the notification and the *parameters* of the sensor. After SPS receiving the task, it will analyze feasibility of the task and then send response to the user.

As the status of the task is confirmed, the tasked sensor will start observing and SOS will collect the observation data from the sensor simultaneously. After the task finished, SPS will receive the notification from Debris Flow Monitoring System, and then inform WNS to notify the user that the task is finished and the observation data can be accessed from SOS.

In our example, *LatestResponseTime* of *SubmitRequestResponse* response is empty. That's because our *SubmitRequestResponse* response is immediate, although in SPS spec 1.0 *LatestResponseTime* is mandatory. And we will further describe it in Chapter 9.

Submit

```
<?xml version="1.0" encoding="UTF-8"?>
<Submit
xmlns="http://www.opengis.net/sps/1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:swe="http://www.opengis.net/swe/1.0"
xsi:schemaLocation="http://www.opengis.net/sps/1.0 http://schemas.opengis.net/sps/1.0.0/spsAll.xsd"
service="SPS" version="1.0.0">
  <notificationTarget>
    <notificationID>c48cceab-c50d-4893-abf5-d135e8aab96b</notificationID>
    <notificationURL>http://ogc.org.tw/wns</notificationURL>
  </notificationTarget>
  <sensorParam>
    <sensorID>urn:ogc:object:Sensor:OGC_TW:8490a2e3-abaf-4a4e-bae2-6875a61a2a5a
    </sensorID>
    <parameters>
      <InputParameter parameterID="zoom">
        <value>
          <swe:Quantity>3</swe:Quantity>
        </value>
      </InputParameter>
      <InputParameter parameterID="pan">
        <value>
          <swe:Quantity>150</swe:Quantity>
        </value>
      </InputParameter>
      <InputParameter parameterID="tilt">
        <value>
          <swe:Quantity>30</swe:Quantity>
        </value>
      </InputParameter>
    </parameters>
  </sensorParam>
</Submit>
```

SubmitRequestResponse

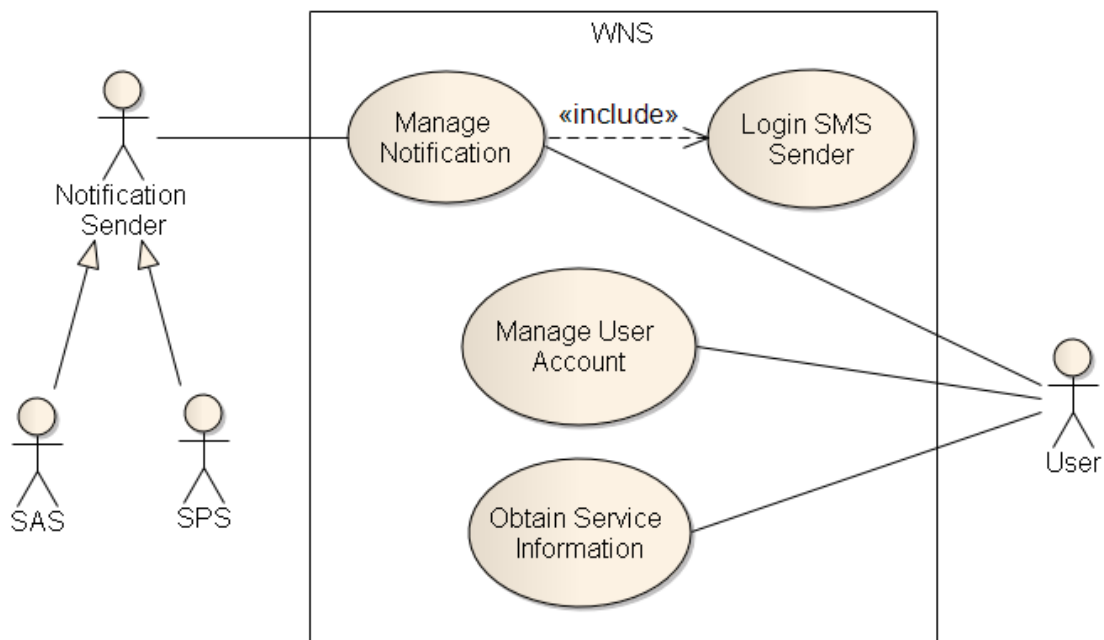
```

<?xml version="1.0" encoding="UTF-8"?>
<SubmitRequestResponse
xmlns:gml="http://www.opengis.net/gml"
xmlns="http://www.opengis.net/sps/1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/sps/1.0 http://schemas.opengis.net/sps/1.0.0/spsAll.xsd">
  <taskID>6ef4d715-a7ed-4254-97ce-3166c856adb6</taskID>
  <status>confirmed</status>
  <LatestResponseTime>
    <gml:TimeInstant>
      <gml:timePosition/>
    </gml:TimeInstant>
  </LatestResponseTime>
</SubmitRequestResponse>

```

8.4 Notification: WNS

Figure 16 is the Use Case Diagram of WNS. This diagram shows that WNS mainly provides three functions *Manage Notification*, *Manage User Account* and *Obtain Service Information*. There are two kinds of users to use these functions. One would be the user and the other one is *Notification Sender*. More specifically, *Notification Sender* can be divided into two parts, SAS and SPS, both of which use WNS to do notification. We will introduce how the user and *Notification Sender* interact with WNS in Figure 17 and Figure 18.

**Figure 16 - Use case diagram of WNS**

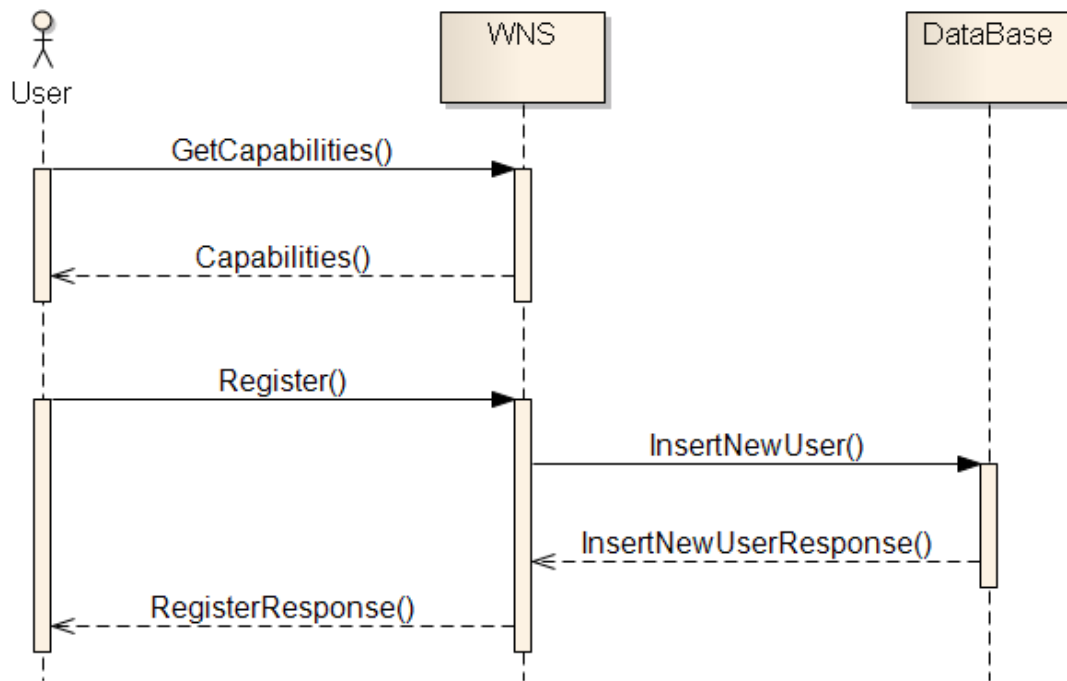


Figure 17 - Sequence diagram of Register

Step 1: Obtain WNS service capabilities (`GetCapabilities`)

To use WNS, the user needs to know what capabilities are provided by WNS. As the major work of WNS is sending notification, the main information the user wants to know is which communication channel can be used to get the notification. And through *GetCapabilities* operation, the user can get all the answers he wants to know. In our following example, we know WNS provides two communication protocols, SMS and Email. And the message will be stored by WNS for ten days.

Capabilities

```

<Contents>
  <SupportedCommunicationProtocols>
    <XMPP>false</XMPP>
    <SMS>true</SMS>
    <Phone>false</Phone>
    <Fax>false</Fax>
    <Email>true</Email>
  </SupportedCommunicationProtocols>
  <MaxTTLOfMessages>P10D</MaxTTLOfMessages>
</Contents>
  
```

Step 2: Register

After knowing the capabilities of WNS, in order to receive the notification, the user needs to *Register* an account in WNS. And it is ok to *Register* either a person or a group in WNS. For personal registration, the user needs to provide his name and the way he wants to receive notifications.

to receive the notification. For group registration, it only needs to provide the *UserID* who wants to join the group. No matter using what kind of registration, **Register** response will only return the *UserID*. From *Capabilities* response example, we know WNS only supports SMS and Email communication. So in the following example, the user can just register both of them.

Register - Single User

```
<?xml version="1.0" encoding="UTF-8"?>
<Register service="WNS" version="1.0.0"
xmlns="http://www.opengis.net/wns/0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wns/0.0 ../wnsAll.xsd">
  <SingleUser>
    <Name>Testuser</Name>
    <CommunicationProtocol>
      <Email>testuser@ogc.org.tw</Email>
      <Email>testuser@ooxx.com</Email>
      <SMS>0912345678</SMS>
    </CommunicationProtocol>
  </SingleUser>
</Register>
```

Register - Multiuser

```
<?xml version="1.0" encoding="UTF-8"?>
<Register service="WNS" version="1.0.0"
xmlns="http://www.opengis.net/wns/0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wns/0.0 ../wnsAll.xsd">
  <MultiUser>
    <UserID>c4ffd611-e444-4bb8-8d54-7d3e9fd9e073</UserID>
    <UserID>7e2285bb-8b72-4d55-bcc4-5280d2491e7e</UserID>
  </MultiUser>
</Register>
```

RegisterResponse

```
<?xml version="1.0" encoding="UTF-8"?>
<RegisterResponse
xmlns="http://www.opengis.net/wns/0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wns/0.0 ../wnsAll.xsd">
  <UserID>c48cceab-c50d-4893-abf5-d135e8aab96b</UserID>
</RegisterResponse>
```

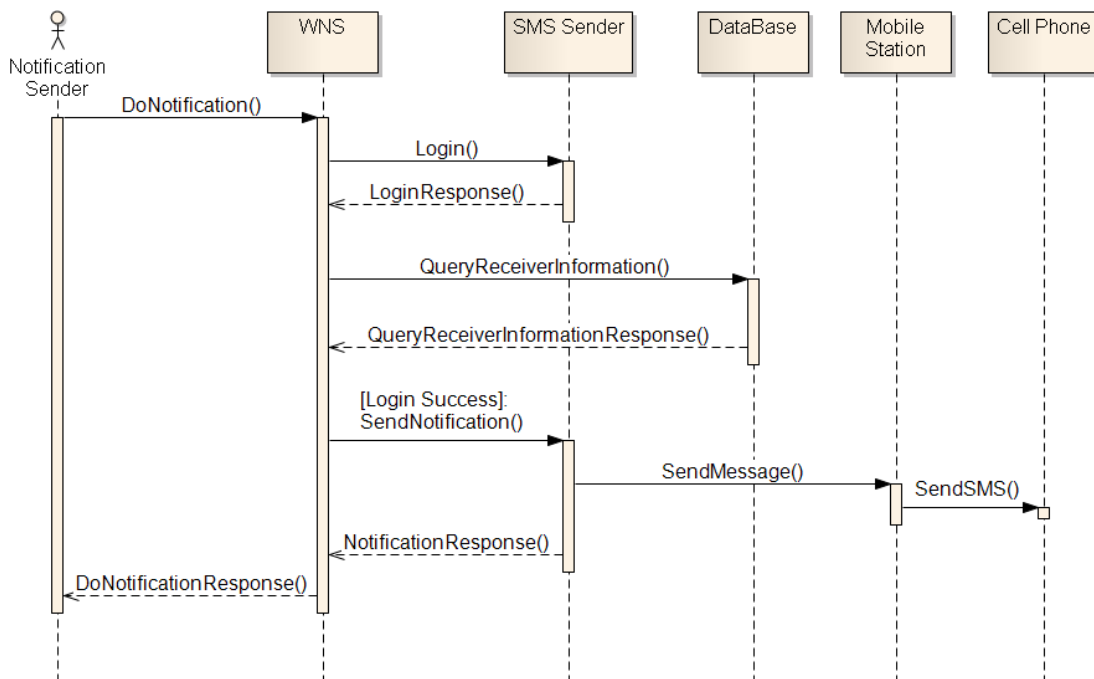



Figure 18 - Sequence diagram of DoNotification

Figure 18 illustrates how Notification Sender interacts with WNS and how WNS deals with the *DoNotification* operation. In this diagram, we use SMS to be our communication protocol.

Step1: DoNotificaion

Notification Sender mainly uses WNS to do notification. In *DoNotification* request, *Notification Sender* provides some information as follows.

-UserID: some the user who want to receive the notification.

-ShortMessage: the information that *Notification Sender* wants to inform the user. In our application, we use the Open GeoSMS format. In the following example, the context of ShortMessage is `GeoSMS/2;(latitude);(longitude);E;[WSN Notification Message] (message) from OGC_TW WSN Data Center` where 'GeoSMS' is the start token for recognition of the short message format; '2' means version 2; latitude and longitude are the GPS coordinates; 'E' represents the Extended format; and the rest is the message to notify the subscribers.

After receiving the *DoNotification* request, first, WNS will *Login* SMS Sender who provides the service to send the SMS. If Login succeeds, WNS then queries DataBase to retrieve the UserID's relevant information. Last, WNS sends *ShortMessage* to the cell phone of the user through SMS Sender.

DoNotification

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<DoNotification service="WNS" version="1.0.0"
xmlns="http://www.opengis.net/wns/0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wns/0.0 ../wnsAll.xsd">
  <UserID>c48cceab-c50d-4893-abf5-d135e8aab96b</UserID>
  <ShortMessage> GeoSMS/2;(latitude);(longitude);E:[WSN Notification Message] (message) from
  OGC_TW WSN Data Center </ShortMessage>
  <Message>
    <NotificationMessage>
      <ServiceDescription>
        <ServiceType>SPS</ServiceType>
        <ServiceTypeVersion>1.0.0</ServiceTypeVersion>
        <ServiceURL>http://ogc.org.tw/sps</ServiceURL>
      </ServiceDescription>
      <Payload>
        <sps:SPSMMessage SPSCorrID="6ef4d715-a7ed-4254-97ce-3166c856adb6"
xmlns:sps="http://www.opengis.net/sps/1.0">
          <sps:StatusInformation>
            <sps:status>Observation data available</sps:status>
          </sps:StatusInformation>
        </sps:SPSMMessage>
      </Payload>
    </NotificationMessage>
  </Message>
</DoNotification>

```

DoNotificationResponse

```

<?xml version="1.0" encoding="UTF-8"?>
<DoNotificationResponse
xmlns="http://www.opengis.net/wns/0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wns/0.0 ../wnsAll.xsd">
  <status>success</status>
</DoNotificationResponse>

```

8.5 Observation: SOS

Figure 19 shows the use case of SOS. The user can use SOS to *Obtain Service Capabilities*, *Manage Sensors*, and *Manage Observations*. *Debris Flow Monitoring System* can use SOS to *Upload Sensor Observation Data*.

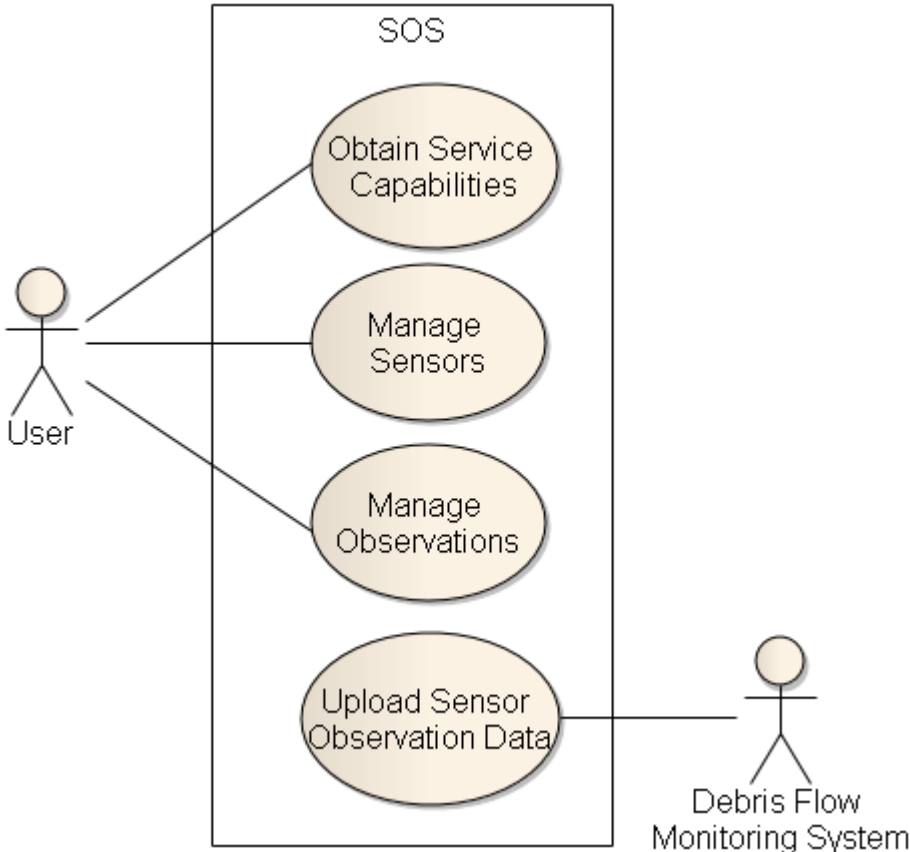


Figure 19 - Use case diagram of SOS

Figure 20 shows how the user gets observations from SOS.

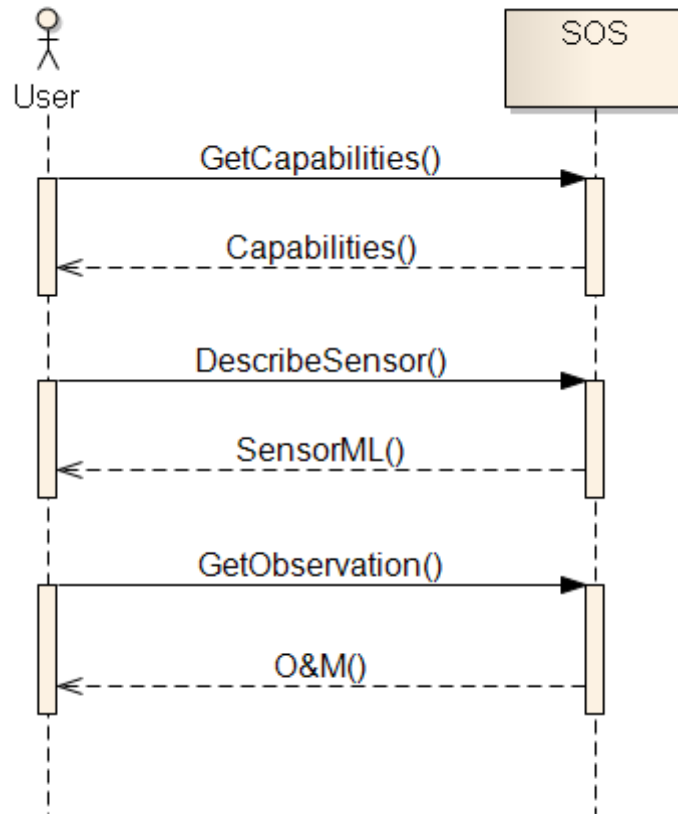


Figure 20 - Sequence diagram of getting observations

Step 1: Get SOS service capabilities

To use an SOS service, the user need to know exactly what observation data the service can provide. This can be achieved through the *GetCapabilities* operation, which allows a client to request and receive back service metadata that describe the abilities of the specific server implementation. The following is an example *GetCapabilities* request XML:

GetCapabilities

```

<?xml version="1.0" encoding="UTF-8"?>
<sos:GetCapabilities xmlns:sos="http://www.opengis.net/sos/1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/sos/1.0 http://schemas.opengis.net/sos/1.0.0/sosAll.xsd"
service="SOS"/>
  
```

The response to a *GetCapabilities* request is a *Capabilities* document:

Capabilities

```

<?xml version="1.0" encoding="UTF-8"?>
<Capabilities ...>
  ...
  <Contents>
    <ObservationOfferingList>
      <ObservationOffering gml:id="offering-7803aae2-ae83-4a29-a6b5-8f2fe66e34e5">
  
```

```

    <gml:name>
      urn:ogc_tw:offering:7803aae2-ae83-4a29-a6b5-8f2fe66e34e5</gml:name>
    <gml:srsName>EPSG:4326</gml:srsName>
    <gml:boundedBy>
      <gml:Null/>
    </gml:boundedBy>
    <time>
      <gml:TimePeriod>
        <gml:beginPosition>2009-01-01T00:00:00+08:00</gml:beginPosition>
        <gml:endPosition indeterminatePosition="unknown"/>
      </gml:TimePeriod>
    </time>
    <procedure xlink:href="urn:ogc:object:Sensor:OGC_TW:bc14680e-c9ad-484b-ae5f-0c97bda5862d"/>
    <observedProperty xlink:href="urn:ogc:def:property:OGC_TW:Rainfall"/>
    <featureOfInterest xlink:href="urn:ogc_tw:feature:Chiu-fen-er-shan"/>
    <responseFormat>text/xml; subtype="om/1.0.0"</responseFormat>
    <resultModel>om:Observation</resultModel>
    <responseMode>inline</responseMode>

  </ObservationOffering>
  ...
</ObservationOfferingList>
</Contents>
</Capabilities>

```

The complete *Capabilities* document is shown in Annex B. In the *Capabilities* document, the *Contents->ObservationOfferingList* element groups related sensor system observations into observation offerings. Our SOS server instance provides four observation offerings for the observed properties: rainfall, ground vibration, wavelet transform of ground vibration, and live image. Figure 21 shows the relationships among feature of interest, observed properties, and procedures in our application.

Figure 21 - Relationships among feature of interest, observed properties, and procedures

Step 2: Get sensor information

With the sensor information in the *Capabilities* document, the user can request observation data using the *DescribeSensor* operation. The following is an example *DescribeSensor* request XML:

```

DescribeSensor
<?xml version="1.0" encoding="UTF-8"?>
<DescribeSensor outputFormat="text/xml;subtype="sensorML/1.0.0";" service="SOS"
version="1.0.0" xmlns="http://www.opengis.net/sos/1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/sos/1.0 http://schemas.opengis.net/sos/1.0.0/sosAll.xsd">
  <procedure>
    urn:ogc:object:Sensor:OGC_TW:8490a2e3-abaf-4a4e-bae2-6875a61a2a5a</procedure>

```

```
</DescribeSensor>
```

In the above example, the *outputFormat* attribute of the *DescribeSensor* element specifies the desired response output format. Here we use the SensorML format. The procedure element specifies the identifier of the sensor whose SensorML the user want to retrieve.

The response to a *DescribeSensor* request is a *SensorML* document, which is show in Annex B.

Step 3: Get observation data

With the observation offering information in the Capabilities document, the user can request observation data using the *GetObservation* operation. In this operation, the user can specify various constraints to filter the retrieved observation data, such as event time, procedure, observed property, feature of interest, etc. The following is an example *GetObservation* request XML:

GetObservation

```
<?xml version="1.0" encoding="UTF-8"?>
<GetObservation xmlns="http://www.opengis.net/sos/1.0" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/sos/1.0 http://schemas.opengis.net/sos/1.0.0/sosAll.xsd"
service="SOS" version="1.0.0">
  <offering>urn:ogc_tw:offering:7803aae2-ae83-4a29-a6b5-8f2fe66e34e5</offering>
  <eventTime>
    <ogc:TM_During>
      <ogc:PropertyName>om:samplingTime</ogc:PropertyName>
      <gml:TimePeriod>
        <gml:beginPosition>2009-04-13T13:00:00+08:00</gml:beginPosition>
        <gml:endPosition>2009-04-13T13:50:00+08:00</gml:endPosition>
      </gml:TimePeriod>
    </ogc:TM_During>
  </eventTime>
  <observedProperty>urn:ogc:def:property:OGC_TW:Rainfall</observedProperty>
  <responseFormat>text/xml; subtype="om/1.0.0"</responseFormat>
</GetObservation>
```

In the above example, the offering element specifies the offering name whose observation data we want to retrieve. This name shall match one of the offering names in the *Contents->ObservationOfferingList->ObservationOffering->gml:name* elements of the *Capabilities* document. The *eventTime* element specifies the sampling time constraint. The *observedProperty* element specifies the property of the target to observe. The *responseFormat* element specifies the output format to be used for the requested data.

The response to a *GetObservation* request is an O&M *Observation*, an element in the Observation substitution group, or an *ObservationCollection*:

Observation

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<om:Observation xmlns:gml="http://www.opengis.net/gml" xmlns:om="http://www.opengis.net/om/1.0"
xmlns:swe="http://www.opengis.net/swe/1.0.1" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/om/1.0 http://schemas.opengis.net/om/1.0.0/om.xsd">
  <gml:name>urn:ogc_tw:observation:276634cb-bde0-4877-9fcb-68ccfb8555ae</gml:name>
  <om:samplingTime>
    <gml:TimePeriod>
      <gml:beginPosition>2009-04-13T13:00:00+08:00</gml:beginPosition>
      <gml:endPosition>2009-04-13T13:50:00+08:00</gml:endPosition>
    </gml:TimePeriod>
  </om:samplingTime>
  <om:procedure
xlink:href="urn:ogc:object:Sensor:OGC_TW:bc14680e-c9ad-484b-ae5f-0c97bda5862d"/>
  <om:observedProperty xlink:href="urn:ogc:def:property:OGC_TW:Rainfall"/>
  <om:featureOfInterest xlink:href="urn:ogc_tw:feature:Chiu-fen-er-shan "/>
  <om:result>
    <swe:DataArray gml:id="RainfallRecords">
      <swe:elementCount>
        <swe:Count>
          <swe:value>6</swe:value>
        </swe:Count>
      </swe:elementCount>
      <swe:elementType name="Components">
        <swe:SimpleDataRecord gml:id="DataDefinition">
          <swe:field name="time">
            <swe:Time definition="urn:ogc:property:time:iso8601"/>
          </swe:field>
          <swe:field name="rainfall">
            <swe:Quantity definition="urn:ogc:def:property:OGC_TW:Rainfall">
              <swe:uom code="mm"/>
            </swe:Quantity>
          </swe:field>
        </swe:SimpleDataRecord>
      </swe:elementType>
      <swe:encoding>
        <swe:TextBlock tokenSeparator="," decimalSeparator="."
          blockSeparator="@@"/>
      </swe:encoding>
      <swe:values>2009-04-13T13:00:00+08:00,0@@@
        2009-04-13T13:10:00+08:00,1@@@
        2009-04-13T13:20:00+08:00,1@@@
        2009-04-13T13:30:00+08:00,7.5@@@
        2009-04-13T13:40:00+08:00,7.5@@@
        2009-04-13T13:50:00+08:00,0
      </swe:values>
    </swe:DataArray>
  </om:result>
</om:Observation>

```

In the above *Observation* document, the *gml:name* element specifies the observation identifier with the format:

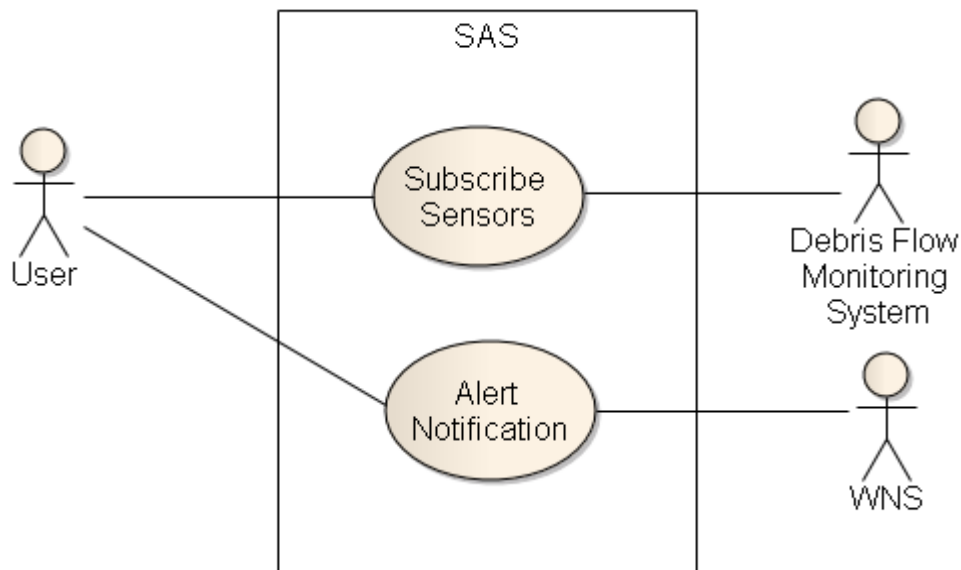
$$\text{urn:ogc_tw:observation:(UUID)}$$

The *samplingTime* element specifies the sampling time period of the observation data. The *om:result* element specifies the format and values of the *observationdata*.

For big-sized observation data such as the geophone raw data, we use the following *out-of-band GetObservation* response mode, which shown as Figure 22.

Figure 22 - Procedure of getting big-sized observation data**8.6 Alert: SAS**

As 8.1 scenario described, the major function of SAS is processing the alert subscription and in this section is going to explain how SAS service operation work in detail. The use case diagram show in Figure 23 and in this diagram the SAS provides two use cases: *Subscribe Sensors* and *Alert Notification*. In *Subscribe Sensors* use case, the user can subscribe sensors that the user want to be noticed when observed data trigger alert events. Another use case is *Alert Notification* and in this case the user has to register WNS services, which describe on previous section, and this registration can provide SAS services ability to send notification message to WNS services. The sequence diagram (Figure 24) will show how two cases are related in detail.

**Figure 23 - Use case of SAS**

SAS sequence diagram of Subscription and Notification (Figure 24) present how the user subscribe sensors and how SAS service provide alert notification to the user. SAS service provides seven operations to the user, the sequence diagram show us in three operations which are *GetCapabilities*, *Subscribe* and *DoNotification*. As you can see, in sequence diagram the user also interacts with WNS service, however WNS service has been described in Section 8.4. Therefore, in this section we only describe SAS service. We will present how SAS service works in 2 steps.

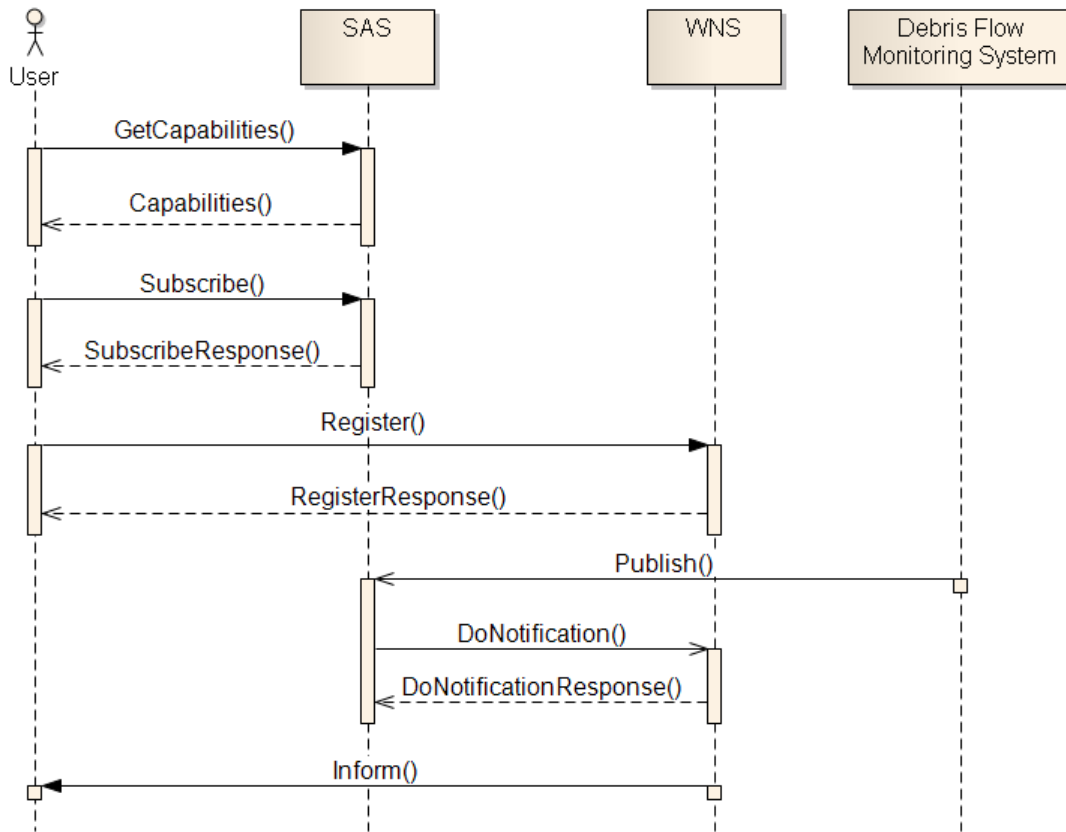


Figure 24 - Sequence diagram of Subscription and Notification

Step 1: Subscription

In order to subscribe sensors, the user needs to know what capabilities SAS service can provide. Therefore, the user can request with *GetCapabilities* operation to SAS service and SAS service shall reply *Capabilities* document to the user, client, to indicate what abilities SAS service can provide. *GetCapabilities* XML document shows below:

```

GetCapabilities
<?xml version="1.0" encoding="UTF-8"?>
<GetCapabilities xmlns="http://www.opengis.net/sas/0.0" xmlns:ows="http://www.opengis.net/ows"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/sas/0.0 ../sasCapabilities.xsd" service="SAS"/>

```

The SAS service responses the *GetCapabilities* operation request with a *Capabilities* XML document. The complete XML document is shown in ANNEX B

```

Capabilities
<?xml version="1.0" encoding="UTF-8"?>
<Capabilities ...>
...
<SubscriptionOffering>

```

```

<SubscriptionOfferingID>urn:ogc_tw:offering:7ba65847-a532-4c5b-8cae-1a13269e5d43
</SubscriptionOfferingID>
<messageStructure>
  <swe:DataBlockDefinition>
    <swe:components name="Rain_Gauge_100_Alert_Message_data_structure" >
      <swe:DataRecord>
        <swe:field name="GeoSMS_Prefix">
          <swe:Text definition="urn:ogc:def:property:OGC_TW:prefix"/>
        </swe:field>
        <swe:field name="GeoSMS_Latitude">
          <swe:Text
            definition="urn:ogc:property:location:EPSG:4326:latitude"/>
        </swe:field>
        <swe:field name="GeoSMS_Longitude">
          <swe:Text
            definition="urn:ogc:property:location:EPSG:4326:longitude"/>
        </swe:field>
        <swe:field name="GeoSMS_Format">
          <swe:Text
            definition="urn:ogc:property:OGC_TW:Format_Type" />
        </swe:field>
        <swe:field name="GeoSMS_Payload">
          <swe:Text definition="urn:ogc:property:OGC_TW:Payload" />
        </swe:field>
      </swe:DataRecord>
    </swe:components>
    <swe:encoding>
      <swe:TextBlock decimalSeparator="." blockSeparator="@@"
        tokenSeparator=";"/>
    </swe:encoding>
  </swe:DataBlockDefinition>
</messageStructure>
<member>
  <SensorID>urn:ogc:object:Sensor:OGC_TW:bc14680e-c9ad-484b-ae5f-
0c97bda5862d</SensorID>
  <Location>
    <swe:Position definition="urn:ogc:def:phenomenon:OGC_TW:Rainfall"
      referenceFrame="urn:ogc:def:crs:EPSG:6.11:4326">
      <swe:location>
        <swe:Vector>
          <swe:coordinate name="Longitude">
            <swe:Quantity>
              <swe:uom code="deg"/>
              <swe:value>120.854216</swe:value>
            </swe:Quantity>
          </swe:coordinate>
          <swe:coordinate name="latitude">
            <swe:Quantity>
              <swe:uom code="deg"/>
              <swe:value>23.950032</swe:value>
            </swe:Quantity>
          </swe:coordinate>
        </swe:Vector>
      </swe:location>
    </swe:Position>
  </Location>
</member>
</SubscriptionOffering>
...
</Capabilities>

```

In *Capabilities* XML document, SAS service provides three kinds of alert notification, Rain fall reaching 100mm (Typhoon alert), Rain fall reaching 250 mm (Debris flow alert, level 1), and Geophone alert (Debris flow alert, level 2). In above gray box is partial *Capabilities* XML document, which shows *SubscriptionOffering->member* and *SubscriptionOffering->messageStructure*. For the *messageStructure* in SAS, we adopt the Open GeoSMS format which is open exchangeable location format for mobile devices for LBS application.

After receiving a *Capabilities* XML document, the user can know which sensors are offering subscription. Therefore, the users can subscribe to SAS service by *Subscribe* request XML document:

Subscribe

```
<?xml version="1.0" encoding="UTF-8"?>
<Subscribe xmlns="http://www.opengis.net/sas/0.0"
xmlns:swe="http://www.opengis.net/swe/1.0"
xmlns:wms="http://www.opengis.net/wms/0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/sas/0.0 ../sasSubscribe.xsd"
service="SAS" version="1.0.0">
  <SensorID>urn:ogc:object:Sensor:OGC_TW:bc14680e-c9ad-484b-ae5f-0c97bda5862d</SensorID>
</Subscribe>
```

In *Subscribe* XML document, we can see the *sensorID* is rain gauge sensor, when we compare to SAS *Capabilities* XML document: *content->SubscriptionOfferingList->SubscriptionOffering->member->SensorID*. SAS shall response *SubscribeResponse* XML document to the user, client, to complete subscription stage.

SubscribeResponse

```
<?xml version="1.0" encoding="UTF-8"?>
<SubscribeResponse SubscriptionID="8d097a48-5fbd-4c93-8b7d-cfa18afc4df8"
expires="2009-05-01T00:00:00Z"
xsi:schemaLocation="http://www.opengis.net/sas/0.0 ../sasSubscribe.xsd"
xmlns="http://www.opengis.net/sas/0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
</SubscribeResponse>
```

Step 2: Alert Notification

In our scenario, Debris flow prediction is provided by Debris Flow Monitoring System that is Feng Chia University GIS center, which is described in chapter 6. Therefore, Alert notice shall be sent by Feng Chia University GIS center to SAS service. In order to implement mechanism of receiving alert notice, we provide a hyper link as below:

sas.aspx?sensorid=[SensorID]&time=[Time]&event=[event]

The parameters of hyper link denote sensor identifier, event time, and what event occurs. After receiving Alert notice, we can use the SensorID to know all sensor information from the SensorML of this sensor. From Alert notice, SAS service is also aware of the time and event. Therefore, SAS service can send *DoNotification* request to WNS. A *DoNotification* request XML document example presents as follows:

DoNotification

```
<?xml version="1.0" encoding="UTF-8"?>
<DoNotification ...>
  <UserID>c48cceab-c50d-4893-abf5-d135e8aab96b</UserID>
  ...
  <Message>
    <NotificationMessage>
      <ServiceDescription>
        <ServiceType>SAS</ServiceType>
        <ServiceTypeVersion>1.0.0</ServiceTypeVersion>
        <ServiceURL>http://ogc.org.tw/sas/</ServiceURL>
      </ServiceDescription>
      <Payload>
        <SASMessage xmlns="http://www.opengis.net/sas/0.0" SubscriptionID="8d097a48-5fbd-4c93-8b7d-cfa18afc4df8">
          <Alert>
            <SensorID>urn:ogc:object:Sensor:OGC_TW:bc14680e-c9ad-484b-ae5f-0c97bda5862d</SensorID>
            <Timestamp>2009-05-24T14:18:22Z</Timestamp>
            <AlertData>GeoSMS/2;2504.8015,N;12133.9766,E;E;Alert:Rain fall reach 100mm from OGC_TW WSN Data Center</AlertData>
          </Alert>
        </SASMessage>
      </Payload>
    </NotificationMessage>
  </Message>
</DoNotification>
```

9 Finding and Outlook

The debris flow monitoring system uses the SWE standard, will change the way of providing the debris flow information data. The traditional way was burned in CD or utilized E-mail way to the user. In the future the user will use the SOS to retrieve the information data via debris flow monitoring system.

Debris flow disaster management needs various type of sensors to monitor the phenomenon of climate, river and ground, however, every sensor has its own characteristics and capabilities, the critical problem SWCB faces is how to integrate and couple these sensors loosely in order to achieve the goal of “plug and play”. Standardization seems to be the optimal solution of this issue. SWCB now is stepping toward to a world of standardized service oriented architecture of monitoring.

In adopting all four SWE services to Debris Flow Monitoring System, There are some findings and, and here we hope these findings can help for future standard.

1. The Submit operation response shall contain a LatestResponseTime element, which specifies that submit response will be sent until LatestResponseTime at latest. However, there is no such time for a non-delayed operation, because the operation responses immediately. So this element should be optional instead of mandatory.
2. The Count data type of SWE Common represents a ‘countable’ property that can be quantified using an integer value, e.g., a number of persons, a number of pixels, etc. However, in many cases, a countable property may need a unit of measure, uom, as the Quantity data type. So we recommend the Count data type to support uom, as the Quantity data type.
3. Using MSN and MMS to communicate with people becomes more and more popular. Therefore, in WNS SupportedCommunicationProtocols, we recommend adding MSN and MMS communication protocols.

ANNEX A

Abstract test suite (to be completed)

ANNEX B

Example of XML documents

B.1 Example of SPS GetCapabilities Request XML document

```
<?xml version="1.0" encoding="UTF-8"?>
<GetCapabilities updateSequence="" service="SPS"
xmlns="http://www.opengis.net/sps/1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/sps/1.0 http://schemas.opengis.net/sps/1.0.0/spsAll.xsd" />
```

B.2 Example of SPS Capabilities Response XML document

```
<?xml version="1.0" encoding="UTF-8"?>
<Capabilities updateSequence="" version="1.0.0"
xsi:schemaLocation="http://www.opengis.net/sps/1.0 http://schemas.opengis.net/sps/1.0.0/spsAll.xsd"
xmlns="http://www.opengis.net/sps/1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:gml="http://www.opengis.net/gml"
xmlns:ows="http://www.opengis.net/ows"
xmlns:xlink="http://www.w3.org/1999/xlink">
  <ows:ServiceIdentification>
    <ows:Title>OGC_TW Debris Flow SPS</ows:Title>
    <ows:Abstract>OGC_TW SPS Server</ows:Abstract>
    <ows:Keywords>
      <ows:Keyword>SPS</ows:Keyword>
      <ows:Keyword>Rain gauge</ows:Keyword>
      <ows:Keyword>CCD</ows:Keyword>
      <ows:Keyword>Geophone</ows:Keyword>
      <ows:Keyword>Debris</ows:Keyword>
      <ows:Keyword>monitoring</ows:Keyword>
      <ows:Keyword>surveillance</ows:Keyword>
    </ows:Keywords>
    <ows:ServiceType codeSpace="http://opengeospatial.net">OGC:SPS</ows:ServiceType>
    <ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
    <ows:Fees>NONE</ows:Fees>
    <ows:AccessConstraints>NONE</ows:AccessConstraints>
  </ows:ServiceIdentification>
  <ows:ServiceProvider>
```

```

<ows:ProviderName>OGC_TW</ows:ProviderName>
<ows:ProviderSite xlink:href="http://ogc.org.tw/sps"/>
<ows:ServiceContact>
  <ows:IndividualName>James Yu</ows:IndividualName>
</ows:ServiceContact>
</ows:ServiceProvider>
<Contents>
  <SensorOfferingList>
    <SensorOffering>
      <AreaOfService>
        <gml:pos>23.950032 120.854216</gml:pos>
      </AreaOfService>
      <Phenomenon>urn:ogc:def:phenomenon:OGC_TW:Rainfall</Phenomenon>
      <SensorDefinition>http://ogc.org.tw/sml/rainSML.xml</SensorDefinition>
      <SensorID>urn:ogc:object:Sensor:OGC_TW:bc14680e-c9ad-484b-ae5f-
        0c97bda5862d</SensorID>
    </SensorOffering>
    <SensorOffering>
      <AreaOfService>
        <gml:pos>23.951081 120.85563</gml:pos>
      </AreaOfService>
      <Phenomenon>urn:ogc:def:phenomenon:OGC_TW:LiveImage</Phenomenon>
      <SensorDefinition>http://ogc.org.tw/sml/ccdSML.xml</SensorDefinition>
      <SensorID>urn:ogc:object:Sensor:OGC_TW:8490a2e3-abaf-4a4e-bae2-
        6875a61a2a5a</SensorID>
    </SensorOffering>
    <SensorOffering>
      <AreaOfService>
        <gml:pos>23.951207 120.85559</gml:pos>
      </AreaOfService>
      <Phenomenon>urn:ogc:def:phenomenon:OGC_TW:GroundVibration</Phenomenon>
      <SensorDefinition>http://ogc.org.tw/sml/geophoneSML.xml</SensorDefinition>
      <SensorID>urn:ogc:object:Sensor:OGC_TW:afe271b3-1cbb-40c9-8d95-
        4e1dac09a373</SensorID>
    </SensorOffering>
  </SensorOfferingList>
  <PhenomenonOfferingList>
    <PhenomenonOffering>
      <Phenomenon>urn:ogc:def:phenomenon:OGC_TW:Rainfall</Phenomenon>
      <SensorID>urn:ogc:object:Sensor:OGC_TW:bc14680e-c9ad-484b-ae5f-
        0c97bda5862d</SensorID>
    </PhenomenonOffering>
    <PhenomenonOffering>

```



```

    <Phenomenon>urn:ogc:def:phenomenon:OGC_TW:LiveImage</Phenomenon>
    <SensorID>urn:ogc:object:Sensor:OGC_TW:8490a2e3-abaf-4a4e-bae2-6875a61a2a
    5a</SensorID>
  </PhenomenonOffering>
<PhenomenonOffering>
  <Phenomenon>urn:ogc:def:phenomenon:OGC_TW:GroundVibration</Phenomenon>
  <SensorID>urn:ogc:object:Sensor:OGC_TW:afe271b3-1cbb-40c9-8d95-
  4e1dac09a373</SensorID>
</PhenomenonOffering>
</PhenomenonOfferingList>
</Contents>
</Capabilities>

```

B.3 Example of SPS DescribeTaskingRequestResponse XML document

```

<?xml version="1.0" encoding="UTF-8"?>
<DescribeTaskingRequestResponse
  xmlns="http://www.opengis.net/sps/1.0"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:swe="http://www.opengis.net/swe/1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/sps/1.0 http://schemas.opengis.net/sps/1.0.0/spsAll.xsd"
  xmlns:xlink="http://www.w3.org/1999/xlink">
  <taskingDescriptor>
    <sensorID>urn:ogc:object:Sensor:OGC_TW:bc14680e-c9ad-484b-ae5f-
    0c97bda5862d</sensorID>
    <InputDescriptor parameterID="rain-sampling-frequency" updateable="false" use="required">
      <gml:description>How often rain gaugu send date back</gml:description>
      <definition>
        <commonData>
          <swe:Quantity definition="urn:ogc:def:parameter:OGC_TW:sampling-
          frequency">
            <swe:uom code="Hz"/>
          </swe:Quantity>
        </commonData>
      </definition>
    </InputDescriptor>
    <InputDescriptor parameterID="threshold" updateable="true" use="required">
      <gml:description>threshold to trigger Geophone</gml:description>
      <definition>
        <commonData>
          <swe:Quantity definition="urn:ogc:def:parameter:OGC_TW:threshold">
            <swe:uom code="mm"/>
          </swe:Quantity>
        </commonData>
      </definition>
    </InputDescriptor>
  </taskingDescriptor>

```

```

        </commonData>
      </definition>
    </InputDescriptor>
  </taskingDescriptor>
</taskingDescriptor>
  <sensorID>urn:ogc:object:Sensor:OGC_TW:8490a2e3-abaf-4a4e-bae2-
6875a61a2a5a</sensorID>
  <InputDescriptor parameterID="zoom" updateable="true" use="optional">
    <gml:description>Zooms the device n steps</gml:description>
    <definition>
      <commonData>
        <swe:Quantity
          definition="urn:ogc:def:parameter:OGC_TW:CCD:AbsoluteZoom">
          <swe:uom code="x"/>
          <swe:constraint>
            <swe:AllowedValues>
              <swe:interval>1 18</swe:interval>
            </swe:AllowedValues>
          </swe:constraint>
        </swe:Quantity>
      </commonData>
    </definition>
  </InputDescriptor>
  <InputDescriptor parameterID="pan" updateable="true" use="optional">
    <gml:description>Pans the device relative to the (0,0) position, Positive means turn right,
    Negative means turn left</gml:description>
    <definition>
      <commonData>
        <swe:Quantity definition="urn:ogc:def:parameter:OGC_TW:CCD:AbsolutePan">
          <swe:uom code="deg"/>
          <swe:constraint>
            <swe:AllowedValues>
              <swe:interval>-170 170</swe:interval>
            </swe:AllowedValues>
          </swe:constraint>
        </swe:Quantity>
      </commonData>
    </definition>
  </InputDescriptor>
  <InputDescriptor parameterID="tilt" updateable="true" use="optional">
    <gml:description>Tilts the device relative to the (0,0) position.</gml:description>
    <definition>
      <commonData>

```

```

        <swe:Quantity
          definition="urn:ogc:def:parameter:OGC_TW:CCD:AbsoluteTilt">
          <swe:uom code="deg"/>
          <swe:constraint>
            <swe:AllowedValues>
              <swe:interval>-30 90</swe:interval>
            </swe:AllowedValues>
          </swe:constraint>
        </swe:Quantity>
      </commonData>
    </definition>
  </InputDescriptor>
</taskingDescriptor>
<taskingDescriptor>
  <sensorID>urn:ogc:object:Sensor:OGC_TW:afe271b3-1cbb-40c9-8d95-
4e1dac09a373</sensorID>
  <InputDescriptor parameterID="geophone-sampling-frequency" updateable="false"
use="required">
    <gml:description>How often geophone to retrieve the data</gml:description>
    <definition>
      <commonData>
        <swe:Time definition="urn:ogc:def:parameter:OGC_TW:sampling-frequency">
          <swe:uom code="Hz"/>
        </swe:Time>
      </commonData>
    </definition>
  </InputDescriptor>
</taskingDescriptor>
</DescribeTaskingRequestResponse>

```

B.4 Example of WNS GetCapabilities Request XML document

```

<?xml version="1.0" encoding="UTF-8"?>
<GetCapabilities service="WNS"
xmlns="http://www.opengis.net/wns/0.0"
xmlns:gml="http://www.opengis.net/gml"
xmlns:ows="http://www.opengis.net/ows"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wns/0.0 ../wnsAll.xsd"/>

```

B.5 Example of WNS Capabilities Response XML document

```

<?xml version="1.0" encoding="UTF-8"?>

```

OGC® 09-082

```
<Capabilities version="1.0.0"
xmlns="http://www.opengis.net/wms/0.0"
xmlns:gml="http://www.opengis.net/gml"
xmlns:ows="http://www.opengis.net/ows"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wms/0.0 ../wmsAll.xsd">
  <ows:ServiceIdentification>
    <ows:Title>OGC_TW WNS</ows:Title>
    <ows:Abstract>OGC_TW WNS</ows:Abstract>
    <ows:Keywords>
      <ows:Keyword>message</ows:Keyword>
      <ows:Keyword>protocol changer</ows:Keyword>
      <ows:Keyword>notification</ows:Keyword>
      <ows:Keyword>communication</ows:Keyword>
    </ows:Keywords>
    <ows:ServiceType codeSpace="http://opengeospatial.net">OGC:WNS</ows:ServiceType>
    <ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
    <ows:Fees>NONE</ows:Fees>
    <ows:AccessConstraints>NONE</ows:AccessConstraints>
  </ows:ServiceIdentification>
  <ows:ServiceProvider>
    <ows:ProviderName>OGC_TW</ows:ProviderName>
    <ows:ProviderSite xlink:href="http://ogc.org.tw/wms"/>
    <ows:ServiceContact>
      <ows:IndividualName>James Yu</ows:IndividualName>
    </ows:ServiceContact>
  </ows:ServiceProvider>
  <ows:OperationsMetadata>
    <ows:Operation name="GetCapabilities">
      <ows:DCP>
        <ows:HTTP>
          <ows:Get xlink:href="http://ogc.org.tw/wms"/>
          <ows:Post xlink:href="http://ogc.org.tw/wms"/>
        </ows:HTTP>
      </ows:DCP>
      <ows:Parameter name="service">
        <ows:Value>WNS</ows:Value>
      </ows:Parameter>
      <ows:Parameter name="version">
        <ows:Value>1.0.0</ows:Value>
      </ows:Parameter>
      <ows:Parameter name="sections">
```

```

    <ows:Value>All</ows:Value>
    <ows:Value>ServiceIdentification</ows:Value>
    <ows:Value>ServiceProvider</ows:Value>
    <ows:Value>OperationsMetadata</ows:Value>
    <ows:Value>Contents</ows:Value>
  </ows:Parameter>
</ows:Operation>
<ows:Operation name="Register">
  <ows:DCP>
    <ows:HTTP>
      <ows:Post xlink:href="http://ogc.org.tw/wms"/>
    </ows:HTTP>
  </ows:DCP>
</ows:Operation>
<ows:Operation name="UpdateSingleUserRegistration">
  <ows:DCP>
    <ows:HTTP>
      <ows:Post xlink:href="http://ogc.org.tw/wms"/>
    </ows:HTTP>
  </ows:DCP>
</ows:Operation>
<ows:Operation name="UpdateMultiUserRegistration">
  <ows:DCP>
    <ows:HTTP>
      <ows:Post xlink:href="http://ogc.org.tw/wms"/>
    </ows:HTTP>
  </ows:DCP>
</ows:Operation>
<ows:Operation name="Unregister">
  <ows:DCP>
    <ows:HTTP>
      <ows:Post xlink:href="http://ogc.org.tw/wms"/>
    </ows:HTTP>
  </ows:DCP>
</ows:Operation>
<ows:Operation name="DoNotification">
  <ows:DCP>
    <ows:HTTP>
      <ows:Post xlink:href="http://ogc.org.tw/wms"/>
    </ows:HTTP>
  </ows:DCP>
</ows:Operation>
</ows:OperationsMetadata>
<Contents>

```

```

    <SupportedCommunicationProtocols>
      <XMPP>>false</XMPP>
      <SMS>>true</SMS>
      <Phone>>false</Phone>
      <Fax>>false</Fax>
      <Email>>true</Email>
    </SupportedCommunicationProtocols>
    <MaxTTLOfMessages>P10D</MaxTTLOfMessages>
  </Contents>
</Capabilities>

```

B.6 Example of SOS Capabilities Response XML document

```

<?xml version="1.0" encoding="UTF-8"?>
<Capabilities xmlns="http://www.opengis.net/sos/1.0" xmlns:gml="http://www.opengis.net/gml"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:swe="http://www.opengis.net/swe/1.0.1"
xmlns:om="http://www.opengis.net/om/1.0" xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:sml="http://www.opengis.net/sensorML/1.0.1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.opengis.net/sos/1.0
http://schemas.opengis.net/sos/1.0.0/sosAll.xsd"
updateSequence="2009-04-28T15:40:00+08:00" version="1.0.0">
  <ows:ServiceIdentification>
    <ows:Title>OGC_TW Debris Flow SOS</ows:Title>
    <ows:Abstract>OGC_TW SOS Server</ows:Abstract>
    <ows:Keywords>
      <ows:Keyword>debris flow</ows:Keyword>
      <ows:Keyword>debris flow alert</ows:Keyword>
    </ows:Keywords>
    <ows:ServiceType
      codeSpace="http://opengeospatial.net">OGC:SOS</ows:ServiceType>
    <ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
    <ows:Fees>NONE</ows:Fees>
    <ows:AccessConstraints>NONE</ows:AccessConstraints>
  </ows:ServiceIdentification>
  <ows:ServiceProvider>
    <ows:ProviderName>OGC_TW</ows:ProviderName>
    <ows:ProviderSite xlink:href="http://ogc.org.tw"/>
    <ows:ServiceContact>
      <ows:IndividualName>James Yu</ows:IndividualName>
    </ows:ServiceContact>
  </ows:ServiceProvider>
  <ows:OperationsMetadata>

```

```

<ows:Operation name="GetCapabilities">
  <ows:DCP>
    <ows:HTTP>
      <ows:Get xlink:href="http://ogc.org.tw/sos/GetCapabilities"/>
      <ows:Post xlink:href="http://ogc.org.tw/sos/GetCapabilities"/>
    </ows:HTTP>
  </ows:DCP>
  <ows:Parameter name="Sections">
    <ows:AllowedValues>
      <ows:Value>ServiceIdentification</ows:Value>
      <ows:Value>ServiceProvider</ows:Value>
      <ows:Value>OperationsMetadata</ows:Value>
      <ows:Value>Contents</ows:Value>
      <ows:Value>All</ows:Value>
    </ows:AllowedValues>
  </ows:Parameter>
</ows:Operation>
<ows:Operation name="DescribeSensor">
  <ows:DCP>
    <ows:HTTP>
      <ows:Post xlink:href="http://ogc.org.tw/sos/DescribeSensor"/>
    </ows:HTTP>
  </ows:DCP>
</ows:Operation>
<ows:Operation name="GetObservation">
  <ows:DCP>
    <ows:HTTP>
      <ows:Post xlink:href="http://ogc.org.tw/sos/GetObservation"/>
    </ows:HTTP>
  </ows:DCP>
</ows:Operation>
<ows:Parameter name="service">
  <ows:AllowedValues>
    <ows:Value>SOS</ows:Value>
  </ows:AllowedValues>
</ows:Parameter>
<ows:Parameter name="version">
  <ows:AllowedValues>
    <ows:Value>1.0.0</ows:Value>
  </ows:AllowedValues>
</ows:Parameter>
</ows:OperationsMetadata>
<Contents>
  <ObservationOfferingList>

```

```

<ObservationOffering gml:id="offering-7803aae2-ae83-4a29-a6b5-8f2fe66e34e5">
  <gml:name>
    urn:ogc_tw:offering:7803aae2-ae83-4a29-a6b5-8f2fe66e34e5</gml:name>
  <gml:srsName>EPSG:4326</gml:srsName>
  <gml:boundedBy>
    <gml:Null/>
  </gml:boundedBy>
  <time>
    <gml:TimePeriod>
      <gml:beginPosition>2009-01-01T00:00:00+08:00</gml:beginPosition>
      <gml:endPosition indeterminatePosition="unknown"/>
    </gml:TimePeriod>
  </time>
  <procedure xlink:href="urn:ogc:object:Sensor:OGC_TW:bc14680e-c9ad-484b-ae5f-0c97bda5862d"/>
  <observedProperty xlink:href="urn:ogc:def:property:OGC_TW:Rainfall"/>
  <featureOfInterest xlink:href="urn:ogc_tw:feature:Chiu-fen-er-shan"/>
  <responseFormat>text/xml; subtype="om/1.0.0"</responseFormat>
  <resultModel>om:Observation</resultModel>
  <responseMode>inline</responseMode>
</ObservationOffering>
<ObservationOffering gml:id="offering-2a7d9a13-87d6-4fcd-a93f-f8a688cec2c2">
  <gml:name>
    urn:ogc_tw:offering:2a7d9a13-87d6-4fcd-a93f-f8a688cec2c2</gml:name>
  <gml:srsName>EPSG:4326</gml:srsName>
  <gml:boundedBy>
    <gml:Null/>
  </gml:boundedBy>
  <time>
    <gml:TimePeriod>
      <gml:beginPosition>2009-01-01T00:00:00+08:00</gml:beginPosition>
      <gml:endPosition indeterminatePosition="unknown"/>
    </gml:TimePeriod>
  </time>
  <procedure xlink:href="urn:ogc:object:Sensor:OGC_TW:afe271b3-1cbb-40c9-8d95-4e1dac09a373"/>
  <observedProperty
    xlink:href="urn:ogc:def:property:OGC_TW:GroundVibration"/>
  <featureOfInterest xlink:href="urn:ogc_tw:feature:Chiu-fen-er-shan"/>
  <responseFormat>text/xml; subtype="om/1.0.0"</responseFormat>
  <resultModel>om:Observation</resultModel>
  <responseMode>out-of-band</responseMode>
</ObservationOffering>

```



```

<ObservationOffering gml:id="offering-42ce45b4-e5af-42ca-aa07-39902d491893">
  <gml:name>
    urn:ogc_tw:offering:42ce45b4-e5af-42ca-aa07-39902d491893</gml:name>
  <gml:srsName>EPSG:4326</gml:srsName>
  <gml:boundedBy>
    <gml:Null/>
  </gml:boundedBy>
  <time>
    <gml:TimePeriod>
      <gml:beginPosition>2009-01-01T00:00:00+08:00</gml:beginPosition>
      <gml:endPosition indeterminatePosition="unknown"/>
    </gml:TimePeriod>
  </time>
  <procedure xlink:href="urn:ogc:object:Sensor:OGC_TW:afe271b3-1cbb-40c9-8d95-4e1dac09a373"/>
  <observedProperty
    xlink:href="urn:ogc:def:property:OGC_TW:GroundVibrationWavelet"/>
  <featureOfInterest xlink:href="urn:ogc_tw:feature:Chiu-fen-er-shan"/>
  <responseFormat>text/xml; subtype="om/1.0.0"</responseFormat>
  <resultModel>om:Observation</resultModel>
  <responseMode>inline</responseMode>
</ObservationOffering>
<ObservationOffering gml:id="offering-ea0a6c7f-dc16-40f7-be0f-f37df7d0ff8f">
  <gml:name>
    urn:ogc_tw:offering:ea0a6c7f-dc16-40f7-be0f-f37df7d0ff8f</gml:name>
  <gml:srsName>EPSG:4326</gml:srsName>
  <gml:boundedBy>
    <gml:Null/>
  </gml:boundedBy>
  <time>
    <gml:TimePeriod>
      <gml:beginPosition>2009-01-01T00:00:00+08:00</gml:beginPosition>
      <gml:endPosition indeterminatePosition="unknown"/>
    </gml:TimePeriod>
  </time>
  <procedure xlink:href="urn:ogc:object:Sensor:OGC_TW:8490a2e3-abaf-4a4e-bae2-6875a61a2a5a"/>
  <observedProperty xlink:href="urn:ogc:def:property:OGC_TW:LiveImage"/>
  <featureOfInterest xlink:href="urn:ogc_tw:feature:Chiu-fen-er-shan"/>
  <responseFormat>text/xml; subtype="om/1.0.0"</responseFormat>
  <resultModel>om:Observation</resultModel>
  <responseMode>inline</responseMode>
</ObservationOffering>
</ObservationOfferingList>

```

B.7 Examples of SensorML document

CCD Camera SensorML document

```

<?xml version="1.0" encoding="utf-16"?>
<SensorML version="1.0.1"
  xmlns="http://www.opengis.net/sensorML/1.0.1"
  xmlns:swe="http://www.opengis.net/swe/1.0.1"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:sml="http://www.opengis.net/sensorML/1.0.1"
  xmlns:gml="http://www.opengis.net/gml"
  xsi:schemaLocation="http://www.opengis.net/sensorML/1.0.1
http://schemas.opengis.net/sensorML/1.0.1/sensorML.xsd">
  <keywords>
    <KeywordList>
      <keyword>CCD</keyword>
      <keyword>Network Camera</keyword>
      <keyword>surveillance</keyword>
    </KeywordList>
  </keywords>
  <identification>
    <IdentifierList>
      <identifier name="longName">
        <Term definition="urn:ogc:def:identifier:OGC_TW:longname">
          <value>OGC_TW AXIS 214 PTZ Network Camera</value>
        </Term>
      </identifier>
      <identifier name="shortName">
        <Term definition="urn:ogc:def:identifier:OGC_TW:shortname">
          <value>OGC_TW CCD</value>
        </Term>
      </identifier>
      <identifier name="modelName">
        <Term definition="urn:ogc:def:identifier:OGC_TW:modelnumber">
          <value>AXIS 214 PTZ</value>
        </Term>
      </identifier>
      <identifier name="manufacturer">
        <Term definition="urn:ogc:def:identifier:OGC_TW:manufacture">
          <value>AXIS</value>
        </Term>
    </IdentifierList>
  </identification>

```

```

        </identifier>
    </IdentifierList>
</identification>
<classification>
    <ClassifierList>
        <classifier name="sensorType">
            <Term>
                <value>Network Camera</value>
            </Term>
        </classifier>
        <classifier name="intendedApplication">
            <Term>
                <value>Debris Flow surveillance and remote monitoring</value>
            </Term>
        </classifier>
    </ClassifierList>
</classification>
<characteristics>
    <swe:DataRecord definition="urn:ogc:def:property:OGC_TW:physicalProperties">
        <swe:field name="physicalProperties">
            <swe:DataRecord>
                <swe:field name="weight">
                    <swe:Quantity definition="urn:ogc:def:property:OGC_TW:weight">
                        <swe:uom code="g"/>
                        <swe:value>1110</swe:value>
                    </swe:Quantity>
                </swe:field>
            </swe:DataRecord>
        </swe:field>
    </swe:DataRecord>
</characteristics>
<characteristics>
    <swe:DataRecord definition="urn:ogc:def:property:OGC_TW:powerRequirement">
        <swe:field name="powerRequirement">
            <swe:QuantityRange definition="urn:ogc:def:property:OGC_TW:power">
                <swe:uom code="V"/>
                <swe:value>11 13</swe:value>
            </swe:QuantityRange>
        </swe:field>
    </swe:DataRecord>
</characteristics>
<capabilities name="operationalLimits">
    <swe:DataRecord definition="urn:ogc:def:property:OGC_TW:operationalLimits">
        <swe:field name="temperatureRange">

```

```

    <swe:QuantityRange
      definition="urn:ogc:def:property:OGC_TW:temperatureRange">
      <swe:uom code="cel"/>
      <swe:value>0 45</swe:value>
    </swe:QuantityRange>
  </swe:field>
  <swe:field name="humidityRange">
    <swe:QuantityRange
      definition="urn:ogc:def:property:OGC_TW:humidityRange">
      <swe:uom code="%"/>
      <swe:value>20 80</swe:value>
    </swe:QuantityRange>
  </swe:field>
</swe:DataRecord>
</capabilities>
<capabilities name="measurementCharacteristics">
  <swe:DataRecord
    definition="urn:ogc:def:property:OGC_TW:measurementCharacteristics">
    <swe:field name="horizontalResolution">
      <swe:QuantityRange
        definition="urn:ogc:def:property:OGC_TW:horizontalResolution">
        <swe:uom code="pixel"/>
        <swe:value>160 704</swe:value>
      </swe:QuantityRange>
    </swe:field>
    <swe:field name="verticalResolution">
      <swe:QuantityRange
        definition="urn:ogc:def:property:OGC_TW:verticalResolution">
        <swe:uom code="pixel"/>
        <swe:value>120 576</swe:value>
      </swe:QuantityRange>
    </swe:field>
    <swe:field name="shutterTime">
      <swe:QuantityRange definition="urn:ogc:def:property:OGC_TW:shutterTime">
        <swe:uom code="s"/>
        <swe:value>0.0001 1</swe:value>
      </swe:QuantityRange>
    </swe:field>
    <swe:field name="panSpeed">
      <swe:Quantity definition="urn:ogc:def:property:OGC_TW:panSpeed">
        <swe:uom code="deg/s"/>
        <swe:value>100</swe:value>
      </swe:Quantity>
  </swe:DataRecord>
</capabilities>

```

```

</swe:field>
<swe:field name="tiltSpeed">
  <swe:Quantity definition="urn:ogc:def:property:OGC_TW:tiltSpeed">
    <swe:uom code="deg/s"/>
    <swe:value>90</swe:value>
  </swe:Quantity>
</swe:field>
</swe:DataRecord>
</capabilities>
<member>
  <System gml:id="ID_8490a2e3-abaf-4a4e-bae2-6875a61a2a5a">
    <position name="actualPosition">
      <swe:Position fixed="false" referenceFrame="urn:ogc:crs:EPSG:4326">
        <swe:location>
          <swe:Vector>
            <swe:coordinate name="longitude">
              <swe:Quantity>
                <swe:uom code="deg"/>
                <swe:value>120.85563</swe:value>
              </swe:Quantity>
            </swe:coordinate>
            <swe:coordinate name="latitude">
              <swe:Quantity>
                <swe:uom code="deg"/>
                <swe:value>23.951081</swe:value>
              </swe:Quantity>
            </swe:coordinate>
          </swe:Vector>
        </swe:location>
      </swe:Position>
    </position>
    <inputs>
      <InputList>
        <input name="DebrisFlowLiveImage">
          <swe:ObservableProperty
            definition="urn:ogc:def:property:OGC_TW:LiveImage"/>
        </input>
      </InputList>
    </inputs>
    <outputs>
      <OutputList>
        <output name="ccd">
          <swe:DataArray
            definition="urn:ogc:def:property:OGC_TW:LiveImage">

```

```

    <swe:elementCount>
      <swe:Count
        definition="urn:ogc:def:property:OGC:numberOfPixels">
        <swe:value>337920</swe:value>
      </swe:Count>
    </swe:elementCount>
    <swe:elementType name="pixel">
      <swe:DataRecord>
        <swe:field name="red">
          <swe:Quantity
            definition="urn:ogc:def:property:OGC:DN"/>
        </swe:field>
        <swe:field name="green">
          <swe:Quantity
            definition="urn:ogc:def:property:OGC:DN"/>
        </swe:field>
        <swe:field name="blue">
          <swe:Quantity
            definition="urn:ogc:def:property:OGC:DN"/>
        </swe:field>
      </swe:DataRecord>
    </swe:elementType>
    <swe:encoding>
      <swe:StandardFormat mimeType="image/jpeg"/>
    </swe:encoding>
  </swe:DataArray>
</output>
</OutputList>
</outputs>
<parameters>
  <ParameterList>
    <parameter name="opticalZoom">
      <swe:Quantity>
        <swe:uom xlink:href="urn:ogc:def:unit:OGC_TW:x"/>
        <swe:constraint>
          <swe:AllowedValues>
            <swe:interval>1 18</swe:interval>
          </swe:AllowedValues>
        </swe:constraint>
      </swe:Quantity>
    </parameter>
    <parameter name="digitalZoom">
      <swe:Quantity>

```

```

    <swe:uom xlink:href="urn:ogc:def:unit:OGC_TW:x"/>
    <swe:constraint>
      <swe:AllowedValues>
        <swe:interval>1 12</swe:interval>
      </swe:AllowedValues>
    </swe:constraint>
  </swe:Quantity>
</parameter>
<parameter name="pan">
  <swe:Quantity>
    <swe:uom code="deg"/>
    <swe:constraint>
      <swe:AllowedValues>
        <swe:interval>-170 170</swe:interval>
      </swe:AllowedValues>
    </swe:constraint>
  </swe:Quantity>
</parameter>
<parameter name="tilt">
  <swe:Quantity>
    <swe:uom code="deg"/>
    <swe:constraint>
      <swe:AllowedValues>
        <swe:interval>-30 90</swe:interval>
      </swe:AllowedValues>
    </swe:constraint>
  </swe:Quantity>
</parameter>
</ParameterList>
</parameters>
</System>
</member>
</SensorML>

```

Rain Gauge SensorML document

```

<?xml version="1.0" encoding="utf-16"?>
<SensorML version="1.0.1"
xmlns="http://www.opengis.net/sensorML/1.0.1"
xmlns:swe="http://www.opengis.net/swe/1.0.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:sml="http://www.opengis.net/sensorML/1.0.1"
xmlns:gml="http://www.opengis.net/gml"
xsi:schemaLocation="http://www.opengis.net/sensorML/1.0.1
http://schemas.opengis.net/sensorML/1.0.1/sensorML.xsd">

```

```

<keywords>
  <KeywordList>
    <keyword>Rain</keyword>
    <keyword>Gauge</keyword>
  </KeywordList>
</keywords>
<identification>
  <IdentifierList>
    <identifier name="longName">
      <Term definition="urn:ogc:def:identifier:OGC_TW:longname">
        <value>OGC_TW Rain Gauge TK1</value>
      </Term>
    </identifier>
    <identifier name="shortName">
      <Term definition="urn:ogc:def:identifier:OGC_TW:shortname">
        <value>OGC_TW Rain Gauge</value>
      </Term>
    </identifier>
    <identifier name="modelName">
      <Term definition="urn:ogc:def:identifier:OGC_TW:modelnumber">
        <value>TK1</value>
      </Term>
    </identifier>
  </IdentifierList>
</identification>
<classification>
  <ClassifierList>
    <classifier name="sensorType">
      <Term>
        <value>Rain Gauge</value>
      </Term>
    </classifier>
    <classifier name="intendedApplication">
      <Term>
        <value>Debris Flow rainfall observing</value>
      </Term>
    </classifier>
  </ClassifierList>
</classification>
<capabilities name="measurementCharacteristics">
  <swe:DataRecord
    definition="urn:ogc:def:property:OGC_TW:measurementCharacteristics">
    <swe:field name="Resolution">
      <swe:Quantity definition="urn:ogc:def:property:OGC_TW:Resolution">
        <swe:uom code="mm"/>
        <swe:value>0.5</swe:value>
      </swe:Quantity>
    </swe:field>
  </swe:DataRecord>
</capabilities>

```



```

</swe:field>
<swe:field name="Frequency">
  <swe:Quantity definition="urn:ogc:def:property:OGC_TW:Frequency">
    <swe:uom code="min"/>
    <swe:value>1</swe:value>
  </swe:Quantity>
</swe:field>
</swe:DataRecord>
</capabilities>
<member>
  <System gml:id="ID_bc14680e-c9ad-484b-ae5f-0c97bda5862d">
    <position name="actualPosition">
      <swe:Position fixed="false" referenceFrame="urn:ogc:crs:EPSG:4326">
        <swe:location>
          <swe:Vector>
            <swe:coordinate name="longitude">
              <swe:Quantity>
                <swe:uom code="deg"/>
                <swe:value>120.854216</swe:value>
              </swe:Quantity>
            </swe:coordinate>
            <swe:coordinate name="latitude">
              <swe:Quantity>
                <swe:uom code="deg"/>
                <swe:value>23.950032</swe:value>
              </swe:Quantity>
            </swe:coordinate>
          </swe:Vector>
        </swe:location>
      </swe:Position>
    </position>
    <inputs>
      <InputList>
        <input name="DebrisFlowRainFall">
          <swe:ObservableProperty
            definition="urn:ogc:def:property:OGC_TW:RainFall"/>
        </input>
      </InputList>
    </inputs>
    <outputs>
      <OutputList>
        <output name="RainFall">
          <swe:Quantity
            definition="urn:ogc:def:property:OGC_TW:RainFall">
            <swe:uom code="mm"/>
          </swe:Quantity>
        </output>
      </OutputList>
    </outputs>
  </System>
</member>

```

```

        </OutputList>
    </outputs>
    <parameters>
        <ParameterList>
            <parameter name="TriggerThreshold">
                <swe:Quantity>
                    <swe:uom code="mm"/>
                </swe:Quantity>
            </parameter>
        </ParameterList>
    </parameters>
</System>
</member>
</SensorML>

```

Geophone SensorML document

```

<?xml version="1.0" encoding="utf-16"?>
<SensorML version="1.0.1"
xmlns="http://www.opengis.net/sensorML/1.0.1"
xmlns:swe="http://www.opengis.net/swe/1.0.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:sml="http://www.opengis.net/sensorML/1.0.1"
xmlns:gml="http://www.opengis.net/gml"
xsi:schemaLocation="http://www.opengis.net/sensorML/1.0.1
http://schemas.opengis.net/sensorML/1.0.1/sensorML.xsd">
    <keywords>
        <KeywordList>
            <keyword>Geophone</keyword>
            <keyword>Vibration</keyword>
        </KeywordList>
    </keywords>
    <identification>
        <IdentifierList>
            <identifier name="longName">
                <Term definition="urn:ogc:def:identifier:OGC_TW:longname">
                    <value>OGC_TW Digital grade long travel Geophone</value>
                </Term>
            </identifier>
            <identifier name="shortName">
                <Term definition="urn:ogc:def:identifier:OGC_TW:shortname">
                    <value>OGC_TW Geophone</value>
                </Term>
            </identifier>
        </IdentifierList>
    </identification>
    <classification>

```

```

<ClassifierList>
  <classifier name="sensorType">
    <Term>
      <value>Geophone</value>
    </Term>
  </classifier>
  <classifier name="intendedApplication">
    <Term>
      <value>Debris Flow Ground Vibration Observing</value>
    </Term>
  </classifier>
</ClassifierList>
</classification>
<capabilities name="measurementCharacteristics">
  <swe:DataRecord
    definition="urn:ogc:def:property:OGC_TW:measurementCharacteristics">
    <swe:field name="frequency">
      <swe:Quantity definition="urn:ogc:def:property:OGC_TW:frequency">
        <swe:uom code="Hz"/>
        <swe:value>500</swe:value>
      </swe:Quantity>
    </swe:field>
  </swe:DataRecord>
</capabilities>
<member>
  <System gml:id="ID_afe271b3-1cbb-40c9-8d95-4e1dac09a373">
    <position name="actualPosition">
      <swe:Position fixed="false" referenceFrame="urn:ogc:crs:EPSG:4326">
        <swe:location>
          <swe:Vector>
            <swe:coordinate name="longitude">
              <swe:Quantity>
                <swe:uom code="deg"/>
                <swe:value>120.85559</swe:value>
              </swe:Quantity>
            </swe:coordinate>
            <swe:coordinate name="latitude">
              <swe:Quantity>
                <swe:uom code="deg"/>
                <swe:value>23.951207</swe:value>
              </swe:Quantity>
            </swe:coordinate>
          </swe:Vector>
        </swe:location>
      </swe:Position>
    </position>
  </member>
</inputs>

```

```

    <InputList>
      <input name="DebrisFlowGroundVibration">
        <swe:ObservableProperty
          definition="urn:ogc:def:property:OGC_TW:GroundVibration"/>
      </input>
    </InputList>
  </inputs>
  <outputs>
    <OutputList>
      <output name="GeophoneRawData">
        <swe:Quantity
          definition="urn:ogc:def:property:OGC_TW:GroundVibration">
          <swe:uom code="cm/s"/>
        </swe:Quantity>
      </output>
    </OutputList>
  </outputs>
</System>
</member>
</SensorML>

```

B.8 Example of SAS Capabilities Response document

```

<?xml version="1.0" encoding="UTF-8"?>
<Capabilities version="1.0.0"
  xmlns="http://www.opengis.net/sas/0.0"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:ows="http://www.opengis.net/ows"
  xmlns:wms="http://www.opengis.net/wms/0.0"
  xmlns:swe="http://www.opengis.net/swe/1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/sas/0.0 ../sasCapabilities.xsd">
  <ows:ServiceIdentification>
    <ows:Title>OGC_TW Debris Flow SAS</ows:Title>
    <ows:Abstract> OGC_TW SAS Server</ows:Abstract>
    <ows:Keywords>
      <ows:Keyword>SAS</ows:Keyword>
      <ows:Keyword>Debris Flow</ows:Keyword>
    </ows:Keywords>
    <ows:ServiceType codeSpace="http://opengeospatial.net">OGC:SOS</ows:ServiceType>
    <ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
    <ows:Fees>NONE</ows:Fees>
    <ows:AccessConstraints>NONE</ows:AccessConstraints>

```

```

</ows:ServiceIdentification>
<ows:ServiceProvider>
  <ows:ProviderName>OGC_TW</ows:ProviderName>
  <ows:ProviderSite xlink:href="http://ogc.org.tw"/>
  <ows:ServiceContact>
    <ows:IndividualName>James Yu</ows:IndividualName>
  </ows:ServiceContact>
</ows:ServiceProvider>
<ows:OperationsMetadata>
  <ows:Operation name="GetCapabilities">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="http://ogc.org.tw/sas/GetCapabilities"/>
        <ows:Post xlink:href="http://ogc.org.tw/sas/GetCapabilities"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
  <ows:Operation name="Subscribe">
    <ows:DCP>
      <ows:HTTP>
        <ows:Post xlink:href="http://ogc.org.tw/sas/Subscribe"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
  <ows:Operation name="RenewSubscription">
    <ows:DCP>
      <ows:HTTP>
        <ows:Post xlink:href="http://ogc.org.tw/sas/RenewSubscription"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
  <ows:Operation name="CancelSubscription">
    <ows:DCP>
      <ows:HTTP>
        <ows:Post xlink:href="http://ogc.org.tw/sas/CancelSubscription"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
  <ows:Operation name="DescribeSensor">
    <ows:DCP>
      <ows:HTTP>
        <ows:Post xlink:href="http://ogc.org.tw/sas/DescribeSensor"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>

```

```

</ows:Operation>
<ows:Operation name="DescribeAlert">
  <ows:DCP>
    <ows:HTTP>
      <ows:Post xlink:href="http://ogc.org.tw/sas/DescribeAlert"/>
    </ows:HTTP>
  </ows:DCP>
</ows:Operation>
</ows:OperationsMetadata>

```

```
<Contents>
```

```

<AcceptAdvertisements>>false</AcceptAdvertisements>
<ReliableCommunicationSupported>>false</ReliableCommunicationSupported>
<SubscriptionOfferingList>
  <SubscriptionOffering>
    <SubscriptionOfferingID>urn:ogc_tw:offering:7ba65847-a532-4c5b-8cae-1a13269e5d43
  </SubscriptionOfferingID>
  <messageStructure>
    <swe:DataBlockDefinition>
      <swe:components name="Rain_Gauge_100_Alert_Message_data_structure" >
        <swe:DataRecord>
          <swe:field name="GeoSMS_Prefix">
            <swe:Text definition="urn:ogc:def:property:OGC_TW:prefix"/>
          </swe:field>
          <swe:field name="GeoSMS_Latitude">
            <swe:Text
              definition="urn:ogc:property:location:EPSG:4326:latitude"/>
          </swe:field>
          <swe:field name="GeoSMS_Longitude">
            <swe:Text
              definition="urn:ogc:property:location:EPSG:4326:longitude"/>
          </swe:field>
          <swe:field name="GeoSMS_Format">
            <swe:Text definition="urn:ogc:property:OGC_TW:Format_Type"
            />
          </swe:field>
          <swe:field name="GeoSMS_Payload">
            <swe:Text definition="urn:ogc:property:OGC_TW:Payload" />
          </swe:field>
        </swe:DataRecord>
      </swe:components>
    <swe:encoding>
      <swe:TextBlock decimalSeparator="." blockSeparator="@@"

```

```
>
```

```

        tokenSeparator=";"/>
      </swe:encoding>
    </swe:DataBlockDefinition>
  </messageStructure>
  <member>
    <SensorID>urn:ogc:object:Sensor:OGC_TW:bc14680e-c9ad-484b-ae5f-
    0c97bda5862d</SensorID>
    <Location>
      <swe:Position definition="urn:ogc:def:phenomenon:OGC_TW:Rainfall"
      referenceFrame="urn:ogc:def:crs:EPSG:6.11:4326">
        <swe:location>
          <swe:Vector>
            <swe:coordinate name="longitude">
              <swe:Quantity>
                <swe:uom code="deg"/>
                <swe:value>120.854216</swe:value>
              </swe:Quantity>
            </swe:coordinate>
            <swe:coordinate name="latitude">
              <swe:Quantity>
                <swe:uom code="deg"/>
                <swe:value>23.950032</swe:value>
              </swe:Quantity>
            </swe:coordinate>
          </swe:Vector>
        </swe:location>
      </swe:Position>
    </Location>
  </member>
</SubscriptionOffering>

<SubscriptionOffering>
  <SubscriptionOfferingID> urn:ogc_tw:offering:c72bcc29-7087-4b5d-8904-f60497a674f7
</SubscriptionOfferingID>
  <messageStructure>
    <swe:DataBlockDefinition>
      <swe:components name="Rain_Gauge_250_Alert_Message_data_structure" >
        <swe:DataRecord>
          <swe:field name="GeoSMS_Prefix">
            <swe:Text definition="urn:ogc:def:property:OGC_TW:prefix"/>
          </swe:field>
          <swe:field name="GeoSMS_Latitude">
            <swe:Text
              definition="urn:ogc:property:location:EPSG:4326:latitude"/>

```

```

        </swe:field>
        <swe:field name="GeoSMS_Longitude">
            <swe:Text
                definition="urn:ogc:property:location:EPSG:4326:longitude"/>
        </swe:field>
        <swe:field name="GeoSMS_Format">
            <swe:Text definition="urn:ogc:property:OGC_TW:Format_Type"
/>

        </swe:field>
        <swe:field name="GeoSMS_Payload">
            <swe:Text definition="urn:ogc:property:OGC_TW:Payload" />
        </swe:field>
    </swe:DataRecord>
</swe:components>
<swe:encoding>
    <swe:TextBlock decimalSeparator="." blockSeparator="@@"
        tokenSeparator=";"/>
</swe:encoding>
</swe:DataBlockDefinition>
</messageStructure>
<member>
    <SensorID>urn:ogc:object:Sensor:OGC_TW:bc14680e-c9ad-484b-ae5f-
0c97bda5862d</SensorID>
    <Location>
        <swe:Position definition="urn:ogc:def:phenomenon:OGC_TW:Rainfall"
referenceFrame="urn:ogc:def:crs:EPSG:6.11:4326">
            <swe:location>
                <swe:Vector>
                    <swe:coordinate name="longitude">
                        <swe:Quantity>
                            <swe:uom code="deg"/>
                            <swe:value>120.854216</swe:value>
                        </swe:Quantity>
                    </swe:coordinate>
                    <swe:coordinate name="latitude">
                        <swe:Quantity>
                            <swe:uom code="deg"/>
                            <swe:value>23.950032</swe:value>
                        </swe:Quantity>
                    </swe:coordinate>
                </swe:Vector>
            </swe:location>
        </swe:Position>

```



```

        </Location>
      </member>
    </SubscriptionOffering>

    <SubscriptionOffering>
      <SubscriptionOfferingID>urn:ogc_tw:offering:03ceefac-a04b-4664-81e4-dcc783cc652e
    </SubscriptionOfferingID>
      <messageStructure>
        <swe:DataBlockDefinition>
          <swe:components name="GeoPhone_Alert_Message_data_structure" >
            <swe:DataRecord>
              <swe:field name="GeoSMS_Prefix">
                <swe:Text definition="urn:ogc:def:property:OGC_TW:prefix"/>
              </swe:field>
              <swe:field name="GeoSMS_Latitude">
                <swe:Text
                  definition="urn:ogc:property:location:EPSG:4326:latitude"/>
              </swe:field>
              <swe:field name="GeoSMS_Longitude">
                <swe:Text
                  definition="urn:ogc:property:location:EPSG:4326:longitude"/>
              </swe:field>
              <swe:field name="GeoSMS_Format">
                <swe:Text definition="urn:ogc:property:OGC_TW:Fornat_Type"
              </swe:field>
              <swe:field name="GeoSMS_Payload">
                <swe:Text definition="urn:ogc:property:OGC_TW:Payload" />
              </swe:field>
            </swe:DataRecord>
          </swe:components>
          <swe:encoding>
            <swe:TextBlock decimalSeparator="." blockSeparator="@@"
              tokenSeparator=";"/>
          </swe:encoding>
        </swe:DataBlockDefinition>
      </messageStructure>
    </member>
    <SensorID>urn:ogc:object:Sensor:OGC_TW:afe271b3-1cbb-40c9-8d95-
    4e1dac09a373</SensorID>
    <Location>
      <swe:Position
        definition="urn:ogc:def:phenomenon:OGC_TW:GroundVibration"
        referenceFrame="urn:ogc:def:crs:EPSG:6.11:4326">

```

```

    <swe:location>
      <swe:Vector>
        <swe:coordinate name="longitude">
          <swe:Quantity>
            <swe:uom code="deg"/>
            <swe:value>120.85559</swe:value>
          </swe:Quantity>
        </swe:coordinate>
        <swe:coordinate name="latitude">
          <swe:Quantity>
            <swe:uom code="deg"/>
            <swe:value>23.951207</swe:value>
          </swe:Quantity>
        </swe:coordinate>
      </swe:Vector>
    </swe:location>
  </swe:Position>
</Location>
</member>
</SubscriptionOffering>
</SubscriptionOfferingList>
</Contents>
<wms:NotificationAbilities>
  <wms:SupportedCommunicationProtocols>
    <wms:XMPP>false</wms:XMPP>
    <wms:SMS>true</wms:SMS>
    <wms:Phone>false</wms:Phone>
    <wms:Fax>false</wms:Fax>
    <wms:Email>true</wms:Email>
    <wms:WSAddressing>false</wms:WSAddressing>
    <wms:WNS>true</wms:WNS>
  </wms:SupportedCommunicationProtocols>
  <wms:SupportedCommunicationFormats>
    <wms:NotificationFormat>basic</wms:NotificationFormat>
  </wms:SupportedCommunicationFormats>
</wms:NotificationAbilities>
</Capabilities>

```