A SOA based debris flow monitoring system

Architecture and proof-of-concept implementation

Lan-Kun Chung, Yao-Min Fang, Tien-Yin Chou, Bing Jean Lee
Geographic Information Systems Research Center
Feng Chia University (FCU)
Taichung, Taiwan

Bastian Baranski
Institute for Geoinformatics
University of Muenster, Germany

Abstract— **Taiwan is located at the collision boundary of the Philippine sea 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 cause extensive disasters, debris flow, the most serious disaster caused by torrential rainfall, lead to very heavy casualties in recent years. There are 17 fixed debris flow monitoring stations and 2 mobile stations deployed in Taiwan. However, the whole architecture was designed in late 2000 and implemented by traditional and proprietary methodologies. Hence, several interoperability issues have been unveiled in the recent years when the needs of interoperability increased. In this study, we propose a whole new and open standards based debris flow monitoring architecture following the Service Oriented Architecture (SOA) paradigm. Relevant Open Geospatial Consortium (OGC) standards (for example Web Processing Service, WPS specification and Sensor Web Enablement, SWE technologies) and advancements from Grid Computing where lead into the proposed architecture. The use of open standards and distributed computing technologies in the proposed architecture enables heterogeneous resources (data, processing and computing power) interoperability. This study also implements an OGC WPS Grid Processing Profile that was developed in the OGC Web Services, Phase 6 (OWS-6) initiative of the OGC Interoperability Program.**

Debris Flow; Service Oriented Architecture; Spatial Data Infrastructure; Open Geospatial Consortium; Web Processing Servic; Sensor Web Enablement; Grid Computing

I. INTRODUCTION

Taiwan is located on the collision boundary of the Philippine Sea 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.

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.

However, without open standard support, it’s not easy to interoperate with a common method to access, manipulate or send a task to a specific sensor, it leads additional cost and without efficiency.OGC Web services(OWS) initiative mainly focuses on the interoperability in geo-spatial territory, hence, debris flow monitoring system becomes a very good test bed of those open standards.

Signifying part of this work is a result of research from within the OGC Web Services, Phase 6 (OWS-6) initiative of the OGC Interoperability Program and will be promoted in further cooperation between the GIS Research Center at Feng Chia University (FCU) and the Institut für Geoinformatik at Universität Münster.

The remainder of this document is structured as follows. Chapter II provides an overview of the fundamental concepts behind the presented work. Chapter III describes the proposed approach of an SOA based Debris Flow monitoring system. Chapter IV provides a more detailed description of the WPS implementation that is used and the developed processes. Chapter V presents the real-world scenario which proves the concept of the previous section. Finally, chapter VI summarizes the key benefits of the proposed architecture and provides an outlook of challenging open research issues.

II. BACKGROUND

The Open Geospatial Consortium (OGC) is a non-profit, international, voluntary consensus standards organization that is leading the development of standards for geospatial and location based services. The OGC specifies several standards for data retrieval, data visualization and data processing. The presented debris flow demo scenario workflow uses OGC Sensor Web Enablement and OGC Web Processing Service technologies for delivering and processing of dynamic sensor data streams. Grid Computing was utilized for increasing the performance of processing large amounts of dynamic geodata. This chapter provides a brief overview of the fundamental concepts behind the presented approach.

A. Sensor Web Enablement

A sensor network is a computer accessible network of spatially distributed devices using sensors to monitor conditions at different locations. The OGC Sensor Web Enablement (SWE) framework of open standards presents many opportunities for adding real-time sensor data to the internet. Within the SWE framework, the enablement of such sensor webs and networks is being pursued through the establishment of several encodings for describing sensors and sensor observations, and through several standard interface definitions for web services.

Beside other specifications from within the OGC SWE framework, the Sensor Observations Service (SOS) is a standard for requesting, filtering, and retrieving observations and sensor system information through a Web Service interface. The SOS is the intermediary between a client and an observation repository or near real-time sensor channel.

B. Web Processing Service

With the advent of the Service Oriented Architecture (SOA) paradigm [3] and general advancements in Web Services technology [15], the geoinformation (GI) world underwent a substantial change from stand-alone applications to distributed service architectures manifested in Spatial Data Infrastructures (SDI) [12]. Existing SDIs are mainly focused on data retrieval, data visualization and data processing [4]. An open standard based SDI mostly supports the retrieval and visualization of data through Web Services. But the data processing was normally done by human actors with more or less proprietary and monolithic Geo Information Systems (GIS). With growing network and processing capacities the web-based processing of dynamic geodata becomes feasible for the first time.

The OGC Web Processing Service (WPS) became an official OGC standard [13] in mid 2007 and is a first but major attempt to address geospatial processes in a standardized way. The WPS specification defines a standardized interface to publish and perform geospatial processes over the internet. Such a process can range from a simple geometric calculation (for example a simple buffer operation) to a complex simulation process (for example a global climate change model).

C. Grid Computing

The term 'Grid Computing' is a diffuse phrase and there are many definitions available. This lack of a sole definition leads to many people working with Grid technology and having different views on what a Grid is. Following the argumentation of Foster [4], the term Grid Computing must be evaluated in „terms of the applications, business value, and scientific results that it delivers, not its architecture“. Foster captured a couple of definitions in his famous three-point-checklist of capabilities, according to which a Grid is a system that:

- coordinates resources that are not subject to centralized control
- is using standard, open, general-purpose protocols and interfaces
- delivers nontrivial Quality of Service (QoS)

From a pragmatically point of view, Grid Computing is a form of distributed computing and a computational Grid is a group of computers that are linked together so that they can be viewed as a single computer. This virtual supercomputer is typically used by scientific communities – for example in high-energy physics at the Large Hadron Collider (LHC) at Cern - to solve problems that were normally infeasible to solve due to computing and data-integration restrictions.

The Open Grid Forum (OGF)[1] is an open community leading the global standardization effort for Grid Computing. It was formed in 2006 in a merger of the Global Grid Forum (GGF) and the Enterprise Grid Alliance (EGA). OGF has a family of specifications that can be applied to accessing distributed computing and data resources. In late 2007 the OGC and the OGF signed a formal Memorandum of Understanding (MoU). This MoU defines particular areas for collaboration, such as the integration of WPS specification with a range of back-end processing environments to enable large-scale processing and the integration of OGC federated catalogs and data repositories with Grid data movement tools.

In the context of integration of Grid Computing or related methods and technologies in geospatial applications only little research has been conducted. Baranski [1] and Laning et al. [10] extended the work of Di et al. [2] and accomplished first experiments using Grid Computing technology for improving processing performance by distributing and parallel execution of processes. Recently, the research project GDI-GRID[2] has started focusing on the integration and processing of geospatial data and services in a grid infrastructure. In particular, it addresses the implementation of security mechanisms and the definition of service interfaces. In [15] Padberg and Kiehle summarize the experiences from GDI-GRID project and examine the conceptual differences between the paradigm of OGC Web Services and Grid computing and provide an integration approach to overcome the gaps between SDIs and Grid infrastructures. The purpose of SEE-GEO[3] project was to provide an access to the EDINA[4] national datacenter hosting geospatial services running on the UK National Grid Service (NGS)[5]. The SLA4D-GRID[6] project starts in June 2009 and focuses on the development of a common software stack for managing and enforcing Service Level Agreements (SLA) in the German grid infrastructure D-GRID via advanced resource reservation mechanisms. The out-sourcing and out-tasking of

1 http://ogf.org/
2 http://www.gdi-grid.de/
3 http://edina.ac.uk/projects/seesaw/index.html
4 http://edina.ac.uk/
5 http://www.grid-support.ac.uk/
6 http://www.sla4d-grid.de/

geospatial processes via an SLA-enabled WPS is one important use-case in this research project. Furthermore, WSRF was proposed by [17] as a mechanism to provide a refactoring of OGC services around a resource-oriented view of data within a SOAP-based web services framework.

III. APPROACH

A. Department of Defense Architecture Framework

The Department of Defense Architecture Framework(DoD AF) 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.

DoDAF v1.5 is a transitional version that responds to the DoD's migration towards Net-Centric Warfare (NCW). It applies essential net-centric concepts in transforming the DoDAF and acknowledges that the advances in enabling technologies – such as services within a Service Oriented Architecture (SOA) – are fundamental to realizing the Department's Net-Centric Vision. Version 1.5 addresses the immediate net-centric architecture development needs of the Department while maintaining backward compatibility with DoDAF v1.0.

There are three views to describe an architecture, system/services view, operational view and technical standards view, respectively. System/services view refers to the related systems, services, and characteristics to operational needs. Operational view identifies what needs to be accomplished and who does it. Technical standards view prescribes standards and conventions. Figure 1 shows the views of DoD AF, operational view(OV), system/services view(SV) and technical standards view(TV), respectively.

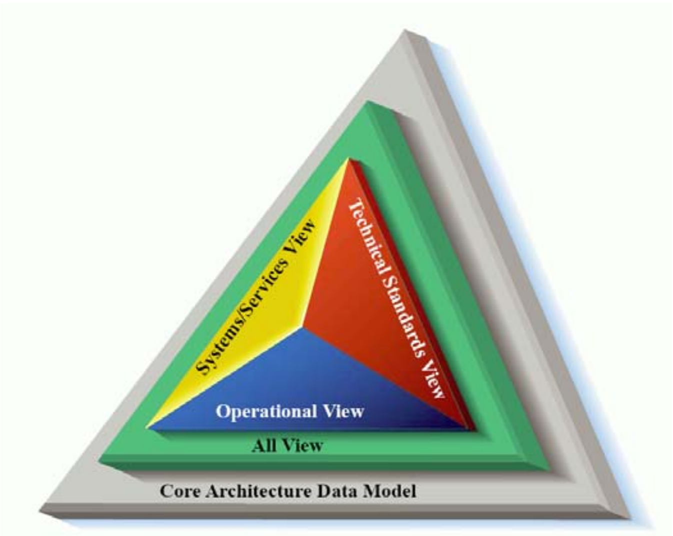

Figure 1. Views of DoD AF

B. Service Oriented Architecture

Service Oriented Architecture (SOA) was proposed by Gartner in 1996, the main concepts of this architecture are loosely coupling and components reusibility, in order to facing a various changes business environment. The general idea of SOA is to drive software to a high flexibility, scalibility, extensibility and plug-and-play architecture without modifying legacy systems "hugely". To conform open standard is also the main concept for adopting SOA into an organization

Figure 2 illustrates the elements of Service Oriented Architecture (SOA) [9]

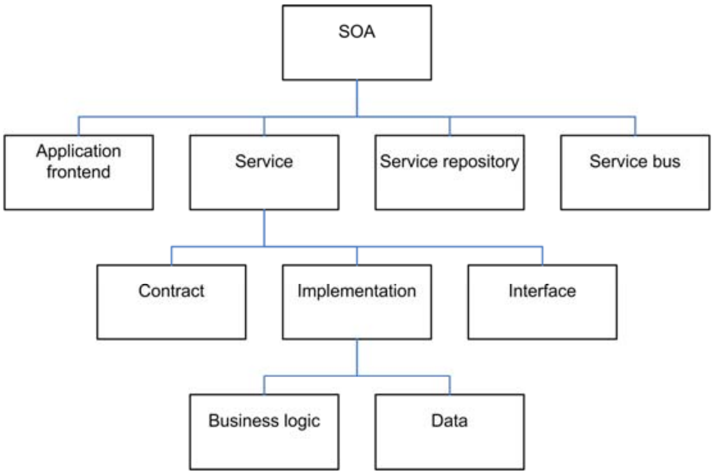

Figure 2 Elements of SOA(by Dirk Krafzig, Karl Banke, and Dirk Slama. Enterprise SOA. Prentice Hall, 2005)

IV. ABSTRACT ARCHITECTURE

Figure 3 illustrates the operational view of debris flow monitoring system by using DoD AF high –level operational concept graphic(OV-1). From satellite to UAV, from fixed monitoring station to field survey are all connected by a standard way. Net-Centric Warfare(NCW) is adopted into this architecture. This architecture is suitable to the whole life cycle of debris flow disaster management, i.e., the mitigation stage which focuses on slope land management, engineering and education works, the preparedness stage which focuses on monitoring technologies development in order to capture and record, of course, to give an early warning to residents, the response stage which centers on emergency response, receives various and heterogeneous sensors data, analyzes and support information to decision maker, the recovery stage which emphasize how to collect post-disaster information and how to manage recovery works.

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 3 shows the general concept of how Debris Flow Monitoring System works in OV-1.

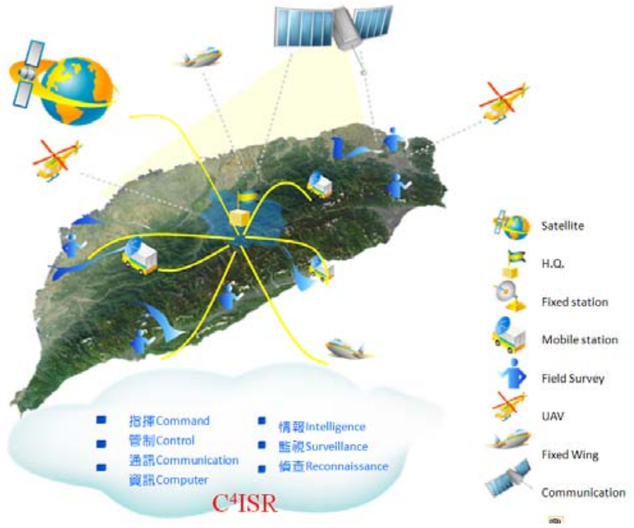

Figure 3. OV-1 of debris flow monitoring system

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 4 illustrates several critical operational nodes in debris flow monitoring system.

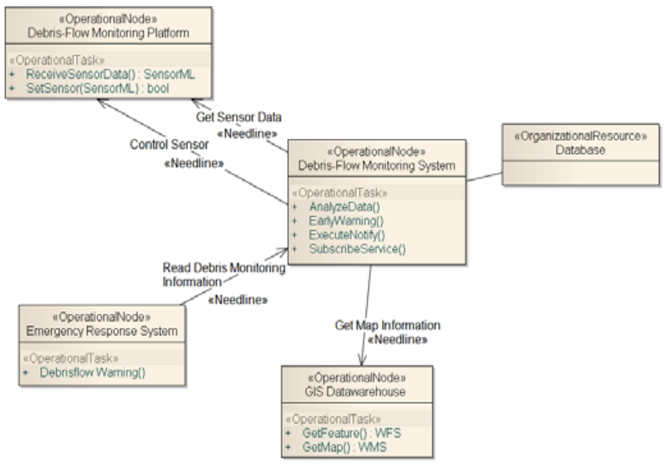

Figure 4. OV-2 of debris flow monitoring system

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 5 illustrates a classical relationship between upstream sensors and downstream decision support activities of debris flow monitoring system

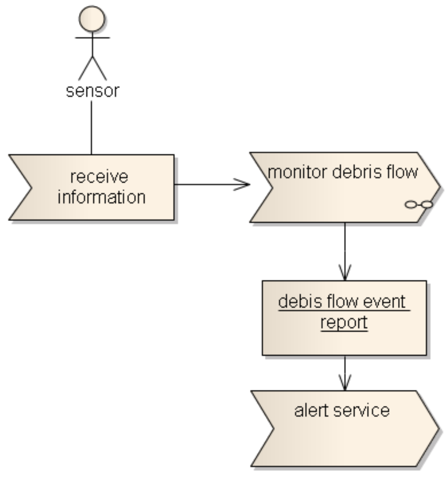

Figure 5. OV-5 of debris flow monitoring system

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 6 shows four main sensors types of a classical debris flow monitoring station.

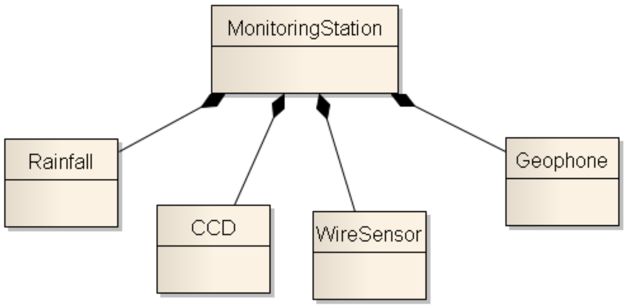

Figure 6. — OV-7 of debris flow monitoring system

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 7 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.

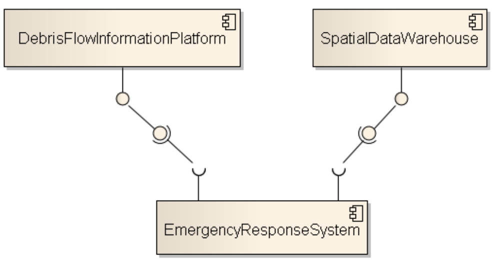

Figure 7. SV-1 for debris flow monitoring system

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 8 is one of system and service view of debris flow monitoring system.

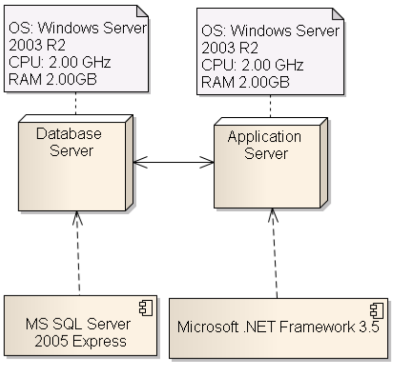

Figure 8. SV-2 for debris flow monitoring system

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 9 shows the relationship between system functions and system data flow in a UML use case diagram.

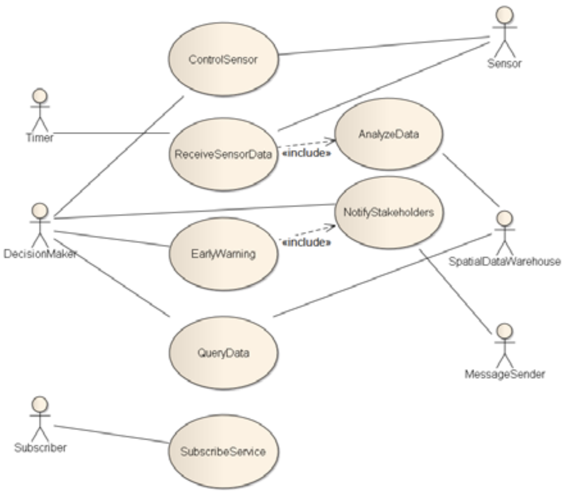

Figure 9. SV-4 of debris flow monitoring system

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

V. IMPLEMENTATION

A. *Sensor Observation Service*

In the proposed SOA based debris flow monitoring architecture a SOS is used to serve the measured rainfall and ground vibration data for later processing through the WPS.

The debris flow monitoring system implements the GetCapabilities, DescribeSensor, and GetObservation three components of operations in SOS. GetCapabilities: This operation allows clients to retrieve service metadata about a specific service instance. No “request” parameter is included, since the element name specifies the specific operation.

DescribeSensor: One method of obtaining metadata that describe the characteristic of an observation procedure (sensor or sensor constellation) is to retrieve it from a catalog. However, a catalog may only contain high-level information about the observable properties, locations, contact information, etc. GetObservation: The GetObservation operation is designed to query a service to retrieve observation data structured according to the Observation and Measure specification. Upon receiving a GetObservation request, a SOS shall either satisfy the request or return an exception report. We use DescribeSensor and rainfall input as follow:

<procedure>urn:fcu:object:feature:Sensor:GIS: **Dacukeng - rainfall** -1</procedure>
Then the rainfall output as follow:
<sml:identifier name="**URN**">
<sml:Term definition="**urn:fcu:def:identifierType:GIS:uniqueID**">
<sml:value>**urn:fcu:object:feature:Sensor:GIS: Dacukeng-Rainfall-1**</sml:value>
<sml:identifier name="**Name**">
<sml:value>**Dacukeng rainfall**</sml:value>
<sml:value>**ID12345**</sml:value>
<sml:identifier name="**Manufacturer**">
<sml:value> **Manufacturer** </sml:value>
<sml:classifier name="**intended Application**">
<sml:value>**rain**</sml:value>
<sml:classifier name="**sensor Type**">
<sml:value>**RainGauge**</sml:value>
<sml:classifier name="**phenomenon**">

```
  <sml:value>rainfall</sml:value>
<sml:System gml:id=" Dacukeng -Rainfall-1">
<sml:position name="rainGaugePosition">
<swe:Position fixed="false"
referenceFrame="urn:ogc:crs:epsg:4326">
<swe:location>
<swe:Vector>
<swe:coordinate name="x">
<swe:Quantity>
  <swe:value>334,004</swe:value>
  </swe:Quantity>
  </swe:coordinate>
<swe:coordinate name="y">
<swe:Quantity>
  <swe:value>2,776,281</swe:value>
<sml:input name="raincount">
  <swe:ObservableProperty
definition="urn:fcu:def:phenomenon:GIS:1.0.0:raincount" />
<sml:output name="RainFall">
<swe:Quantity definition="urn:fcu:def:phenomenon:GIS:1.0.
0:rainfall">
  <swe:uom code="mm" />
```


B. *Web Processing Service*

In the proposed SOA based debris flow monitoring architecture a Grid Computing enabled WPS is used to analyze and process real-time rainfall and ground vibration data.

Two Grid encapsulating [14] processes where developed in the proposed demo scenario:

- Rainfall Interpolation

  The Rainfall Interpolation process implements an Inverse Distance Weighting (IDW) algorithm and interpolates the rainfall data measured by the deployed rain gauges.

- Ground Vibration Analysis

  The Ground Vibration Analysis process implements a Wavelet Transformation algorithm and calculates if the ground vibration measured reaches a specific threshold.

Both WPS processes receive the sensor measurements through the SOS describe in the previous chapter. Whereas the WPS server itself is deployed at the Institute for Geoinformatics at University of Münster, the two processes for analyzing and processing the real-time rainfall and ground vibration data are performed at Jülich Supercomputing Centre (JSC)[7] at Research Center Jülich (FZJ). Figure 10 shows the debris flow monitoring system scenario workflow.

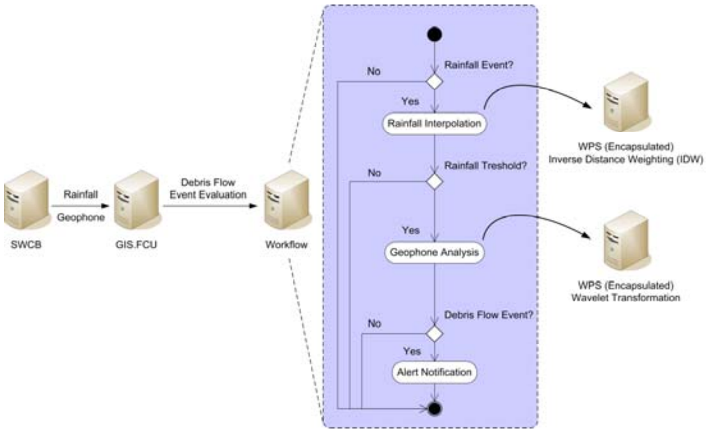

Figure 10. Debris flow monitoring system scenario workflow

The developed Grid Computing enabled WPS is based on the 52°North[8] implementation of WPS 1.0.0 specification. It is public available under the GNU General Public License (GPL) at the 52°North open source repository. For remotely invoking applications in a Grid Computing infrastructure the open source grid middleware UNICORE[9] is utilized. An architecture overview of the WPS implementation is shown in Figure 11.錯誤! 找不到參照來源。

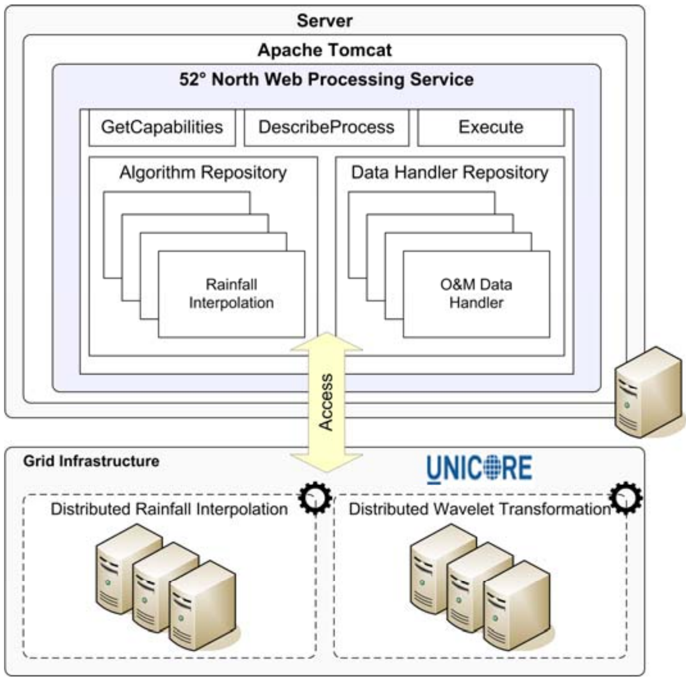

Figure 11. Architecture of Grid-enabled 52°North WPS

For executing distributable processes, the 52° North WPS divides processing tasks into smaller sub-tasks by splitting up the input data into smaller chunks (following a classic Divide-and-Conquer approach and inspired by the Google Map/Reduce framework). Then these chunks and necessary application binaries will be distributed upon several computational nodes in the Grid. The number of computation

[7] http://www.fz-juelich.de/jsc/index.php?index=3

[8] http://52north.org/

[9] http://www.unicore.eu

nodes depends on the complexity of the process and can be either adjusted manually (for example dynamically through the WPS interface or statically via a configuration file) or assessed automatically. After preparing each computational node by submitting the required input data and application binaries onto each node, the process tasks will be executed in parallel. During this execution phase, the CPU at the WPS server side is idle and is waiting for process results or other incoming WPS Execute requests. When all computational nodes have performed the execution successfully, the WPS server fetches all result datasets and merges them together.

Geoprocesses are often considered to be very complex and therefore the effort to divide and conquer them is in most cases very high [7] and in some cases the Divide-and-Conquer approach is not applicable for geospatial problems. However, if geospatial processing tasks are indivisible, the Grid Computing infrastructure can still be used to find the laziest node for processing.

VI. DEMO SCENARIO

If it begins to rain, decision maker can start to analyze if the debris flow will be happened or not. The client send the request to a pre-defined workflow in a BPEL engine then execute the workflow. The workflow sends the request to sensor database and asks for rain gauge and geophone data. If it's a rainfall event, workflow will invoke rainfall interpolation WPS grid service, this WPS-grid service will start to compute parallelly in order to accelerate the computing speed. If the rainfall reach the threshold after computing, work flow will invoke geophone analysis WPS grid service and start computing. If this is a debris flow event, workflow will send the message to client and notify the decision maker.

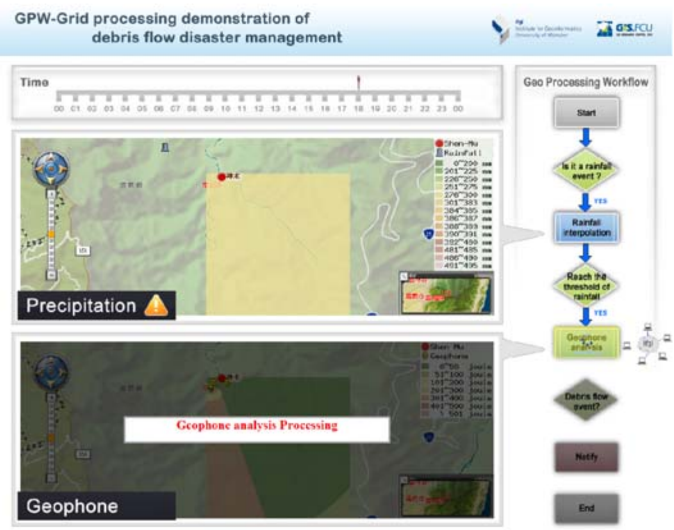

Figure 12. Screenshot of OWS-6 demo

Figure 12 illustrates the demonstration of debris flow disaster management, a workflow orchestrated by a BPEL 2.0 compliant engine is on the right hand side, two maps shows the result of precipitation and geophone on the left hand side, respectively. A debris flow event will be analyzed by this pre-defined workflow, decision maker will receive an alert after processing.

VII. CONCLUSION

A. *Summary*

SOA is a good paradigm for heterogenous data sources integration and orchestration especially those data sources are distributed at different geographic location and with different interfaces. This study implement OGC WPS, OGC SOS,OASIS WS-BPEL and OGSA specification to demonstrate how SOA helps debris flow disaster management and illustrates its performance when grid computing is integrated into this architecture.

SOS has three mandatory "core" operations: GetObservation, DescribeSensor, and GetCapabilities. SOS provided access to observations from sensors and sensor systems in a standard way that is consistent for all sensor systems including remote, in-situ, fixed and mobile debris flow monitoring stations.

B. *Outlook*

Independent of the introduced SOA based debris flow architecture the authors identified general challenging open research issues concerning the integration of distributed computing resources into geospatial applications.

First, the integration and enforcement of Service Level Agreements (SLA) in SDIs. The availability of OWS and OWS-based SDI's rapidly grow for the last decade and they'll supposable play a major role in emerging business models. Monitoring the performance of loosely coupled Web Services in a distributed infrastructure and the ability to react quickly on service quality fluctuations is overall an essential skill for service consumers and service providers. Particularly with regard to the Infrastructure for Spatial Information in the European Community (INSPIRE) directive, in which several performance criteria requirements for geospatial network services are defined. Existing OWS specifications and standards do not support any kind of service quality enforcement functionality. Therefore, normally a formal contract between a service consumer and a service provider - the Service Level Agreement (SLA) - is negotiated [11]. The SLA formalizes a business relationship and enables contractual parties to measure, manage and enforce certain Quality of Service (QoS) guarantees. Attaching SLA functionality to OWS will pose a great advancement for future business models and Grid Computing is a promising for actively enforcing the promised service quality goals from within the SLAs.

Second, the utilization of Cloud Computing methods to enable new and promising business models. The emerging term Cloud Computing as one of the latest trends in the mainstream IT overlaps with some concepts of distributed and Grid Computing [6]. It uses a cloud image to represent the internet or other large networking infrastructures. In essence, the term Cloud Computing collects a family of well known and established methods and technologies and describes a paradigm of outsourcing specific tasks to a scalable infrastructure and consequently enabling new business models. The Cloud Computing paradigm is promising for geospatial applications to enable new and promising business models.

References

[1] Baranski, B. (2008). Grid Computing Enabled Web Processing Service. GI-Days 2008, Münster, Germany

[2] Di, L., Chen, A., Yang W., & Zhao, P. (2003). The Integration of Grid Technology with OGC Web Services (OWS) in NWGISS for NASA EOS Data . GGF8 & HPDC12. 24 – 27 June at Seattle.

[3] Erl, T. (2005). Service-Oriented Architecture : Concepts, Technology, and Design. Upper Saddle River, Prentice Hall PTR.

[4] Foster I., Kesselman C. (1998), „The Grid: Blueprint for a New Computing Infrastructure.", Morgan Kaufmann Publishers

[5] Foster I. (2000), „What is the Grid? A Three Point Checklist"

[6] Hartig. K (2008) What is Cloud Computing? The cloud is a virtualization of resources that maintains and manages itself. .NET Developers Journal, SYS-CON Media.

[7] Hazel, T., Toma, L., Vahrenhold, J. & Wickremesinghe, R. (2008). Terracost: Computing least-cost-path surfaces for massive grid terrains. J. Exp. Algorithmics 12 (Jun. 2008), 1-31.

[8] Kiehle, C., Greve, K. & C. Heier (2007). Requirements for Next Generation Spatial Data Infrastructures - Standardized Web Based Geoprocessing and Web Service Orchestration. In: Transactions in GIS. 11(6), p. 819-834.

[9] Krafzig, D., Banke, K., and Slama, D. 2004 Enterprise Soa: Service-Oriented Architecture Best Practices (The Coad Series). Prentice Hall PTR.

[10] Lanig, S., Schilling, A., Stollberg, B. & Zipf, A. (2008). Towards Standards-based Processing of Digital Elevation Models for Grid Computing through Web Processing Service (WPS). The 2008 International Conference on Computational Science and its Applications (ICCSA2008), Perugia, Italy.

[11] John Lee and Ron Ben-Natan. Integrating Service Level Agreements: Optimizing Your OSS for SLA Delivery. John Wiley & Sons, Inc., New York, NY, USA, 2002.

[12] Masser, Ian (2005). GIS worlds : Creating spatial data infrastructures. 1st ed. Redlands, California: ESRI Press.

[13] OGC 05-007r7, OpenGIS® Web Processing Service Specification

[14] OGC 09-041, OGC® OWS-6 WPS Grid Processing Profile Engineering Report (under review)

[15] Padberg, A. and Kiehle, C. (2009). Towards a grid-enabled SDI: Matching the paradigms of OGC Web Services and Grid Computing. International Journal of Spatial Data Infrastructures Research, Special Issue GSDI-11 (under review)

[16] Weerawarana, Sanjiva (2006). Web services platform architecture: SOAP, WSDL, WS-policy, WS-addressing, WS-BPEL, WS-reliable messaging and more. 4. printing ed. Upper Saddle River, N.J. u.a.: Prentice Hall/PTR.

[17] Woolf, A (2006). Wrappers, portlets, resource-orientation and OGC in Earth-System Science Grids, Grid ad-hoc, OGC TC Edinburgh, June 2006