

Article

Applying OGC Sensor Web Enablement Standards to Develop a TDR Multi-Functional Measurement Model

Chih-Chung Chung ^{1,*} , Chih-Yuan Huang ², Chih-Ray Guan ¹ and Ji-Hao Jian ¹

¹ Dept. of Civil Engineering, National Central University, 300 Zhongda, Rd., Zhongli Dist., Taoyuan 320, Taiwan; harry4558719@gmail.com (C.-R.G.); apple900d@gmail.com (J.-H.J.)

² Center for Space and Remote Sensing Research, National Central University, 300 Zhongda, Rd., Zhongli Dist., Taoyuan 320, Taiwan; cyhuang@csr.sr.ncu.edu.tw

* Correspondence: ccchung@ncu.edu.tw; Tel.: +886-3-422-7151 (ext. 34120); Fax: +886-3-425-2960

Received: 13 August 2019; Accepted: 18 September 2019; Published: 20 September 2019



Abstract: Time-domain reflectometry (TDR) is considered as a passive monitoring technique which reveals multi-functions, such as water level, bridge scour, landslide, and suspended sediment concentration (SSC), based on a single TDR device via multiplexing and related algorithms. The current platform for revealing TDR analysis and interpreted observations, however, is complex to access, thus a coherent data model and format for TDR heterogeneous data exchange is useful and necessary. To enhance the interoperability of TDR information, this research aims at standardizing the TDR data based on the Open Geospatial Consortium (OGC) Sensor Web Enablement (SWE) standards. To be specific, this study proposes a TDR sensor description model and an observation model based on the Sensor Model Language (SensorML) and Observation and Measurement (O&M) standards. In addition, a middleware was developed to translate existing TDR information to a Sensor Observation Service (SOS) web service. Overall, by standardizing TDR data with the OGC SWE open standards, relevant information for disaster management can be effectively and efficiently integrated in an interoperable manner.

Keywords: Time-domain reflectometry (TDR); interoperability; Sensor Observation Service (SOS)

1. Introduction

The technology of Internet-connected networks has brought evolutionary changes from academic research to daily life in recent decades. Data collection is automated and convenient now, and related management has a profound impact on monitoring in civil engineering and disaster prevention domains. Time-domain reflectometry (TDR) technology, as a one-dimension cable radar, is one of the passive monitoring techniques featured by interpreting the reflected electromagnetic (EM) wave due to the different environmental changes. TDR has been widely used in geotechnical, structural, and hydrological engineering monitoring, and even in agriculture [1,2]. The TDR monitoring applications include bridge pier scour [3–6], suspended sediment concentration (SSC) [7], water pressure or level [8–10], soil moisture content [11–15], soil electrical conductivity (EC) [16–18], slope sliding [19–22], and mine subsidence [23]. These references to case studies and specific challenges encountered in geotechnical and soil science applications of TDR demonstrate what kinds of engineering decisions have been or may be supported by TDR measurements. In addition, a TDR monitoring system not only contains the monitoring equipment at the site, but also cooperates with the data transmission module (such as Electronic Industry Association Recommended Standard 232 (RS232), General Packet Radio Service (GPRS), Wireless Local Area Networking (WLAN), Low Power, Wide Area (LPWA) networking (LoRaWAN), or Narrow Band-Internet of Things (NB-IoT)), which can automatically

retrieve information from in-situ to a server [24]. This kind of real-time data is valuable for disaster early warning. Furthermore, the cause or the failure mechanisms of disasters can be discovered by analyzing the historical data. Based on the effective data interchange for aforementioned TDR implementations, previous works have put considerable effort into developing some early data interchange models for TDR and related instrument data [25,26]. The current TDR analysis with the related parameters, however, is still complex to access. Thus, well-considered models for interoperability are useful and necessary.

To improve the interoperability of TDR data, there are many feasible options from Internet of Things (IoT) and Sensor Web standard-definition bodies. As the IoT has received much attention recently, private companies have defined various proprietary data models and service protocols. This heterogeneity issue is usually solved with hub solutions that communicate with different ecosystems via customized connectors [27]. However, the hub approach, which is achieved by implementing different connectors corresponding to different Sensor Web and Internet of Things (SW-IoT) ecosystems, needs to use customized connectors for different protocols, leading to a significant development cost.

On the other hand, following open standards is an attractive approach to achieve data interoperability. Among the potential open-standard solutions for TDR data, the Open Geospatial Consortium (OGC) Sensor Web Enablement (SWE) provides a set of standards that are commonly seen as a comprehensive solution for the Sensor Web on the application layer [28]. For example, the Observation and Measurement (O&M) standard defines a general but complete data model for sensor observations [29], and the Sensor Model Language (SensorML) provides a flexible framework and schema to describe various types of sensor metadata based on an Extensible Markup Language (XML) derivative [30]. XML is one of the markup languages that defines rules for encoding documents in a format that is both human-readable and machine-readable. In addition, the Sensor Observation Service (SOS) standard is defined to provide a coherent service protocol for sensor data sharing [31].

Although the OGC SWE aims at providing an interoperable framework for general sensor data, specific domain data models within the SWE ecosystem still need to be defined to really achieve interoperability in each domain. For example, WaterML 2.0 is one of the standard information models for the representation of water observational data, which allows the exchange of such datasets across information systems [32]. Based on the existing OGC standards, it attempts being an interoperable exchange format that may be re-used to address a range of exchange requirements.

In addition, Sofos et al. [33] implemented the OGC O&M and SOS in describing the land surveying measurement model, especially focusing on the Total Station/Positioning System (TPS), and XML request documents were subsequently developed. Stasch et al. [34] developed coupling sensor observation services and web processing services for online geoprocessing in water dam monitoring by implementing the standardized XML-based interface and lightweight Representational State Transfer (REST) APIs, where REST is a software architectural style that defines constraints to be used for creating Web services. Web services that conform to the REST architectural style are called RESTful Web services (RWS) and provide interoperability between computer systems on the Internet [35]. Based on the aforementioned successful cases, this study aims at proposing a domain solution for integrating the TDR data with the OGC SWE ecosystem.

Furthermore, one thing should be mentioned is that the OGC SWE proposes a new service protocol standard, the SensorThings API [36], to share sensor data in a REST service. The SensorThings API applies the Java Script Object Notation (JSON) format to represent sensor data, where JSON is considered as a low-overhead alternative to XML [37]. Although SOS is based on Simple Object Access Protocol (SOAP) XML, which is heavier than the JSON format, XML can represent the basic and fundamental content by permitting strong typing, user-defined types, predefined tags, and a formal structure for describing the TDR model.

Overall, this study attempts to propose a recommended model for TDR monitoring data under the OGC SWE standards to improve data interoperability. Afterward, a case study is then introduced

by integrating a TDR monitoring system with an SOS service. Then, an OGC Web Processing Service (WPS) service supporting landslide monitoring and pre-warning strategies is demonstrated to reveal an interoperable manner by using TDR.

2. TDR Basics

TDR technology was developed from the detection of cable deformation and fault in the telecommunications industry in 1950. Today's TDR technology is used widely in geotechnical engineering, agricultural engineering, environmental science, electronic engineering, and earth engineering [1]. The operating principle of TDR is similar to that of radar, except that the EM pulse travels through a cable rather than free space. A TDR device, as shown in Figure 1, includes an EM step-pulse generator, a signal sampler with or without an oscilloscope. The step-pulse generator generates an EM step-pulse into the transmission cable and the waveguide via a multiplexer, and the transmission cable causes the EM wave reflection signal to be displayed on the oscilloscope due to the discontinuity of electrical characteristic impedance (e.g., changes of the geometry or internal material of the cable/waveguide).

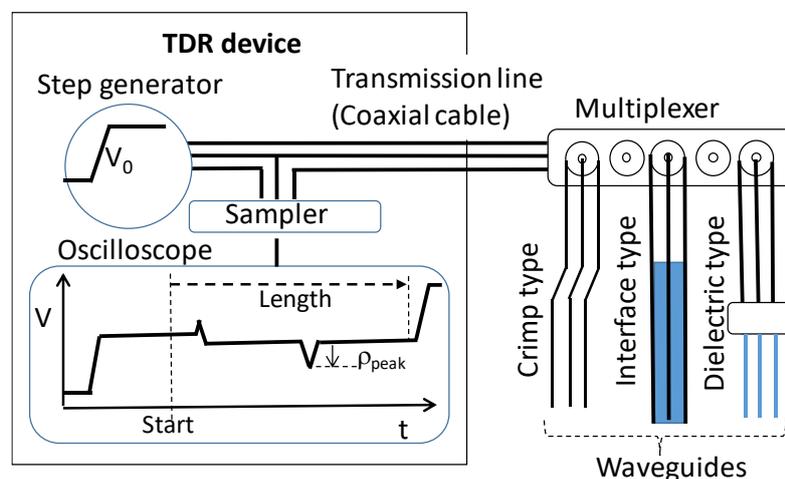


Figure 1. Time-domain reflectometry (TDR) schematic and related sensing waveguides.

Based on the different design of the waveguide, TDR is applied to various monitoring applications, such as a TDR landslide monitoring system with multi-functions as shown in Figure 2 [38]. Generally, TDR can be divided into three types in engineering monitoring applications. The first one is Crimp Type, where the waveguide is the cable itself for landslide monitoring. The sensing waveguide is an off-the-shelf coaxial transmission line and is mainly placed into a borehole with the grout in a soil/rock slope. When the slope slides, it changes the geometry of the coaxial transmission line, causing the reflection spike, indicating the corresponding depth and degree of sliding, as shown in Figure 1. The reflected waveform with spikes is then transmitted and analyzed via wire/wireless communication for landslide early warning. Chung et al. [39] proposed a solution using a time-differential method between reference and measured TDR waveforms as well as a noise-based criterion to automatically identify the level and location of the waveform spike due to the sliding surface.

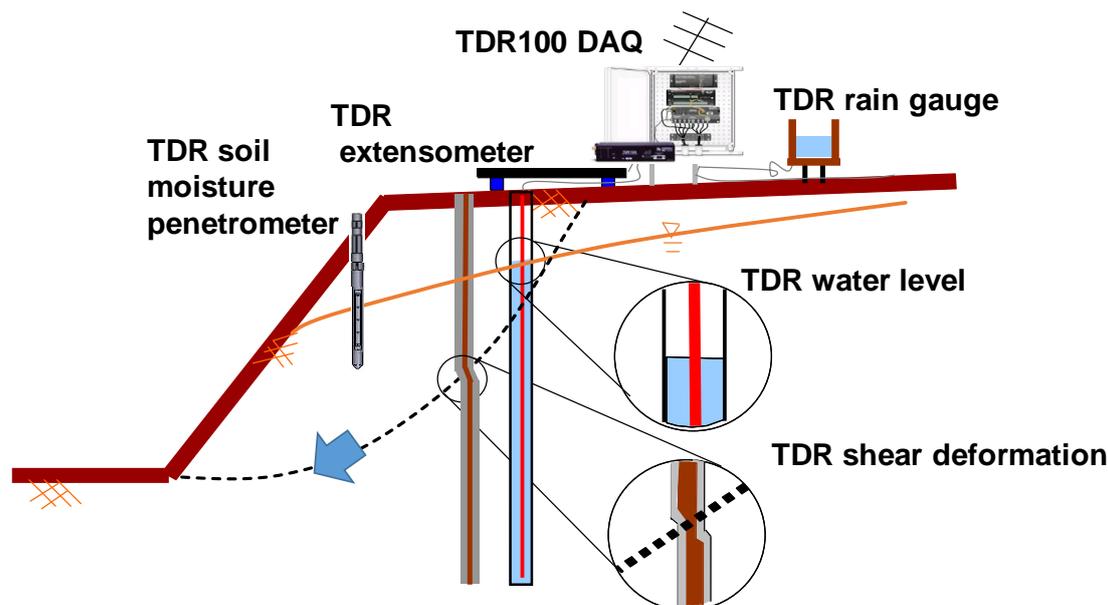


Figure 2. TDR landslide monitoring system with multi-functions, adapted from [22].

Although TDR landslide monitoring has high temporal and spatial resolution as well as high measurement sensitivity of detecting the thin shear surface, the landslide dip direction cannot be determined with a single borehole. Nevertheless, the quantification of shear displacement of landslide is case-dependent [18]. Thus Chung et al. [39] discovered that using time variations of the TDR reflected spikes at certain depths can be practical for landslide early warning and evacuation of residents nearby.

The second one is Interface Type, which measures the reflection at the interface between different materials, such as bridge pier scour [3–6], water pressure and water level [8–10]. The third one is Dielectric Type. This type is commonly used to quantify different dielectric properties of the medium, such as the dielectric permittivity for soil moisture [11–15], EC [16–18], and SSC [7].

In current operation of the TDR device, such as TDR100/200 (Campbell Scientific) or TDR3000 (Sympuls Aachen), several settings are required to capture the reflected EM waveforms. For example, START POSITION is the beginning of an observation window of a captured waveform for TDR3000, as shown in Figure 1; TIME STEP is the sampling time interval between two samples of a captured waveform; NO. of SAMPLES is the number of data samples of a captured waveform; CHANNEL is the channel of the multiplexer. Furthermore, AVG means the stacking number of TDR waveforms for noise reduction. These parameters are required for TDR manipulation and as references of the subsequent TDR waveform analysis among the aforementioned three measurement types.

3. OGC-SWE Initiative

OGC defines the SWE standard that can be applied to different fields with sensors, transducers, and sensor data repositories discoverable, accessible and useable via the Web, and that can be read and controlled remotely. Based on the concept of SWE, the relevant specifications [28] can be divided into information models and schema, and services through the coding descriptions, supporting the Web service agreement and exchanging various service information, as shown in Figure 3. SWE Common defines modules that exchange sensor-related data with each other under the SWE framework [28]. SensorML provides the framework of describing metadata of sensors and systems in an XML schema [30], meanwhile, the O&M defines models of the sensor measurement and observation in an XML schema as well [40]. This paper firstly involves the SensorML to reveal the sensor (waveguide) characteristics and process of data acquisition using TDR.

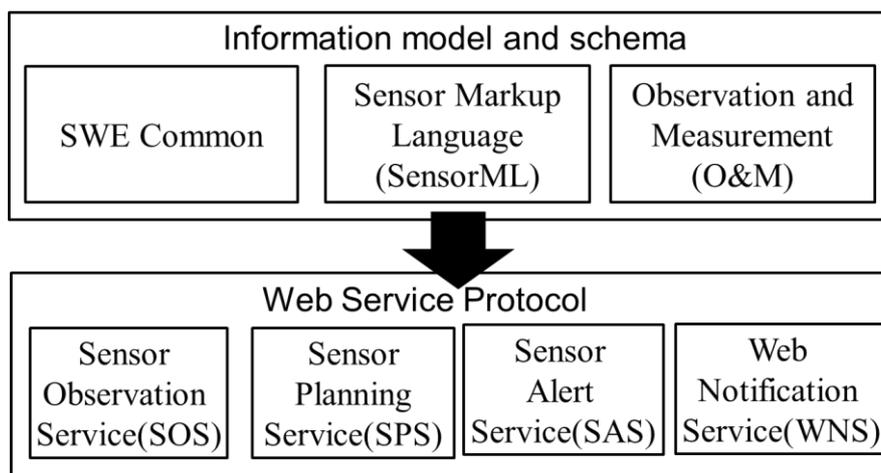


Figure 3. Framework of Sensor Web Enablement (SWE).

Web Notification Service (WNS) uses non-synchronous transmission to assist forward notifications of other service sensor information in SWE to the client, and SOS provides a standard to define the service interface for requesting, filtering, and retrieving the sensor system information and sensor-related data observation [31]. Please refer to Section 3.3 for details of SOS.

Sensor Planning Service (SPS) is the service interface for sensor work planning. Users can search for tasks through the SPS service interface, and then the SPS assigns tasks to the sensors. When the sensor task is completed, the SPS uses WNS to notify the user and places the data into the SOS. Finally, Sensor Alert Service (SAS) provides a subscription sensor to generate data, and users subscribe the sensor information through the query condition on SAS. When the sensor generates data and transmits it to the SAS, the SAS refers the alert setting conditions by users and then notifies the user through the WNS transmission.

3.1. Sensor Model Language (SensorML)

SensorML, which is one of several implementation standards produced under OGC's SWE activity as previously mentioned, focuses on the process of measurement and observation, rather than on sensor hardware. Thus, SensorML provides a robust and semantically-tied means of defining processes and processing components associated with the measurement and post-measurement transformation of observations, including the sensors, actuators, as well as computational processes applied pre- and post-measurement.

The standard of SensorML v2.0 proposes sensor metadata models and an XML implementation. The conformance rules for the XML implementation are based on XML validation using XML Schema representation of SensorML together with processing constraints expressed using Schematron assertions and reports, in which Schematron is well-known as a structural based validation language, and it is as an alternative to existing grammar based approaches [30].

3.2. Observation and Measurement Standard (O&M)

O&M standard defines model and related XML schema to describe information of sensor observations and measurements. The O&M standard was published as ISO 19156: 2011 [41]. The standard was originally implemented in XML format, as well as the JSON format standard which was released in 2015 [42]. The standard can be extended be implemented with RESTful transmission method [32], which describes a process from the act of observation to results across a wide variety of application domains. Based on the above advantages, O&M is a core standard in SWE and is a key module for exchanging information. Core class diagrams of O&M are then defined as shown in Figure 4. OM_Observation, as the fundamental class, supports five attributes, six associations, and four conditional restrictions. The main attributes contain:

1. parameter (optional): use NamedValue to describe. This is for arbitrary event-specific parameters, e.g., instrument settings.
2. phenomenonTime (mandatory): use TM_Object to describe. This is the time that the result applies to the feature of interest, or the acquisition duration.
3. resultTime (mandatory): Use TM_Instant to describe the format. This is the completion time of the entire observation program, analysis, and simulation.
4. validTime (optional): use TM_Period to describe. This is the time period during which the result is intended to be used.
5. resultQuality (optional): Use DQ_Element (ISO 19115-1: 2014) [43] to describe the quality of the result, which can be a supplementary description of the procedure.

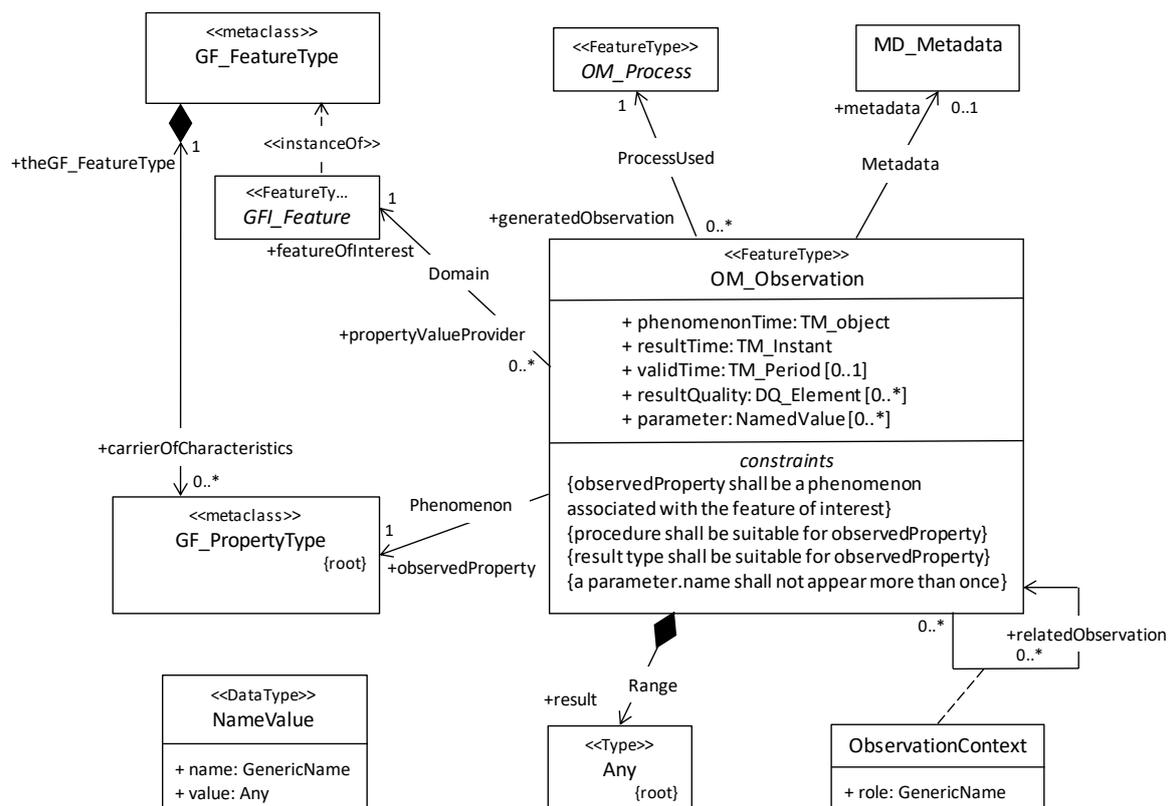


Figure 4. Core class diagram of Observation and Measurement O&M conceptual model [29].

The associated description contains:

1. metadata: This describes general properties such as the data identifier, the downlink and archiving information to the observation.
2. procedure: This is referenced as the **OM_Process** class in the procedure. Mainly defines the description of factors affecting the observations, such as platform/instrument/sensor and event operators and algorithms used for the acquisition and the acquisition parameters [29]. **OM_Process** does not require definition of attribute and is an abstraction category.
3. result: This is the final result of observation.
4. featureOfInterest: Describe real-world measurement objects that is being observed such as rivers, bridges, regions, etc. There is no limit in determining the observed target to its description.
5. observedProperty: Define the properties of the featureOfInterest that are being observed or acquired by the procedure.
6. relatedObservation: This description represents zero or more **OM_Observations**, where their relationship is essential to provide more context.

The restrictions include:

1. The observedProperty is considered as the property or phenomenon whose value is described or estimated through observation and must be related to the featureOfInterest.
2. The procedure and result must be consistent with the observedProperty.
3. The name of parameter must be unique.

3.3. Sensor Observation Service(SOS)

SOS mainly improves the heterogeneous sensing data format and provides access to observations from sensors and sensor systems by proposing a standard service interface. SOS is based on SOAP (Simple Object Access Protocol) and key/value pair transmission methods, where SOAP is an XML-based messaging protocol specification for exchanging structured information in the implementation of web services [44]. SOS uses standard service interface and standard description format to represent sensor observations and measurements in an interoperable manner [31].

SOS began in 2007 with the release of the SOS 1.0 standard [45], defining three sensor service standards in Core, Enhanced, and Transactional. Core provides core observations. Core contains GetCapabilities, DescribeSensor, and GetObservation. GetCapabilities provides detailed information about this SOS server such as service unit, network interface communication protocol and registration of sensor information. DescribeSensor provides the query function of the registered sensors or sensor systems represented by the SensorML specification. GetObservation provides the service access to sensor observations with the filtering through ranges of time and space domains, Procedure, featureOfInterest, and observedProperty. Transactional imports information such as sensors and observations. Enhanced provides an augmented query method in addition to Core. In the past, the standards and guidelines that refer to modeling of the observation procedure were proposed as SOS v1.0. Song et al. conducted urban environmental monitoring through low-cost sensors (including temperature, humidity, air quality, and brightness) and proposed data open methods and formats [46]. Sorg and Kunkel developed an access standard for aeronautical or raster data in SOS v1.0 and revealed a data module which can provide effective access to metadata and databases [47].

This study follows the SOS v2.0 [31], which is based on four specifications: Core, Transactional, Result Handling, and Enhanced Operations. Result Handling is a new category of SOS 2.0. Result Handling can import by self-developing template to reduce transmission bandwidth. In addition, SOS v2.0 strictly follows OGC Web Service Common Specification [48], SWE Service Model 2.0 [49], Filter Encoding [50] and O&M [41] Pearlman et al. proposed the monitoring data representation format for the Oceans of Tomorrow (OoT) project [51] based on SOS v2.0. Additionally, Sofos et al. defined the land survey data module using SOS v2.0. This model data group realized multi-point monitoring station spatial information query [33].

Currently, SOS implementations are available via 52° North SOS [52], Map Server SOS [53] and Deegree SOS [54]. Among these, limited SOS functions (GetCapabilities, DescribeSensor, and GetObservation) are provided in Map Server SOS and Deegree SOS, while the 52° North SOS supports all SOS features including Transactional and Enhanced Operations. Thus, this study applies the 52° North SOS in the experiments.

4. Definition of TDR Model Profile for SOS

Based on the aforementioned TDR monitoring technique (e.g., water level, SSC, and deformation) and the SOS-related methods that have been included in the previous sections, this section explores TDR model profile for SOS. First, the fundamental framework the TDR model profile is composed of is as follows:

1. Procedure, Offering, and observableProperty by providing the registration methods of TDR Project info, Station, and Measured Property;
2. Class diagram of TDR OM_Process conceptual model;

3. Insert sensor with XML request.

All these points follow the definition of O&M in the feature of interest-observed property-process-result. Following sections will discuss the allocation method and finally implementation of the architecture of the data module through the SOS v2.0.

4.1. Registration of Procedure, Offering and observableProperty

Due to the consideration of the required information of the project which consists of the monitoring stations and subsidiary TDR monitoring sensor types, this study establishes a TDR registration method in SOS. Figure 5 first depicts the definition of the SOS, consisting of Project Info, Station, and Measured Property. One or more monitoring stations may be included in a proposed project, and one or more TDR sensor types, such as SSC, water level, and landslide monitoring, would be included in a single monitoring station. The measurement by each TDR sensor type in a monitoring station is unique, thus Project Info, Station and Measured Property are referenced as the SOS identification codes of Procedure, Offering and observableProperty, respectively.

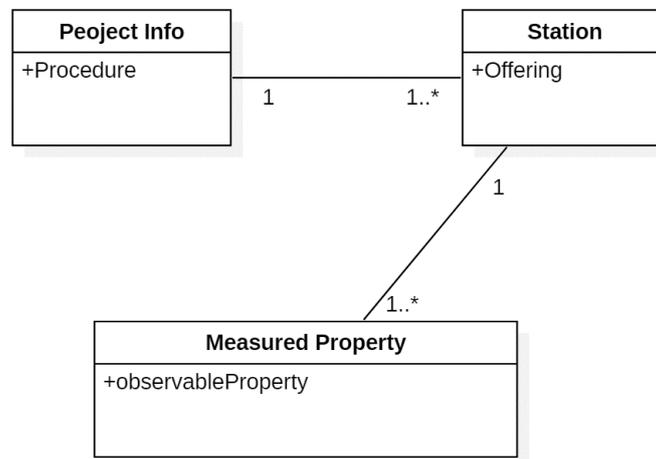


Figure 5. Definition of the relationship diagram of the SOS registration code for TDR.

4.2. Class Diagram of TDR OM_Process Conceptual Model

This study uses OM_Process to define basic information about TDR monitoring processes, such as SSC (TDR_SSC_Process), water level (TDR_WL_Process) and conductivity (TDR_EC_Process), as shown in Figure 6. OM_ComplexObservation is further applied to the crimp type (TDR_Deformation_Process) due to the acquired waveform feature of TDR landslide application. These classes inherit the OM_Process class, and each monitoring process includes monitoring device (OnsiteDevice), operator information (Operator), station information (StationInfo), and TDR device setting information (TDRSetting) as previously mentioned. The following defines the parameters in detail:

1. Field device information (TDR_Device): device name (Name) and device model (Model), such as TDR3000 (Sympuls Aachen) or TDR100/200 (Campbell Scientific).
2. Operator Information (TDR_Operator): Name, Email, and the phone number of Contact.
3. Station information (TDR_Deformation_Info, TDR_SSC_Info, TDR_WL_Info, TDR_EC_Info): According to different TDR monitoring methods, different station information is proposed. First, the crimp type monitoring (TDR_Deformation_Info), as referred to in Figure 6, includes station name (Title), waveform filtering method (FilterType), depth of slope (ElevationOfStart), waveform effective range (StartPoint and EndPoint), depth conversion coefficient (DepthInterval), reference waveform (RefWave) and sliding threshold value (Threshold). Second, suspended sediment concentration monitoring (TDR_SSC_Info) contains the station name (Title), warning threshold (Threshold) and probe depth (Depth). Third, water level monitoring (TDR_WL_Info)

contains the station name (Title), the water level elevation (Elevation) and the warning threshold (Threshold). Final, EC monitoring (TDR_EC_Info) contains the station name (Title) and the warning threshold (Threshold).

4. TDR host setting information (TDR_Device_Setting): TDR mainframe is divided into two models, TDR3000 (Sympuls Aachen) or TDR100/200 (Campbell Scientific). TDR100/200 setting includes captured window range (Start and Length), total number of waveform data (NumOfData), and average number of waveforms for stacking (Avg). TDR3000 setting includes start point of waveform (Start), time interval (TimeInterval), total number of waveform data (NumOfData), and average number of waveforms for stacking (Avg).

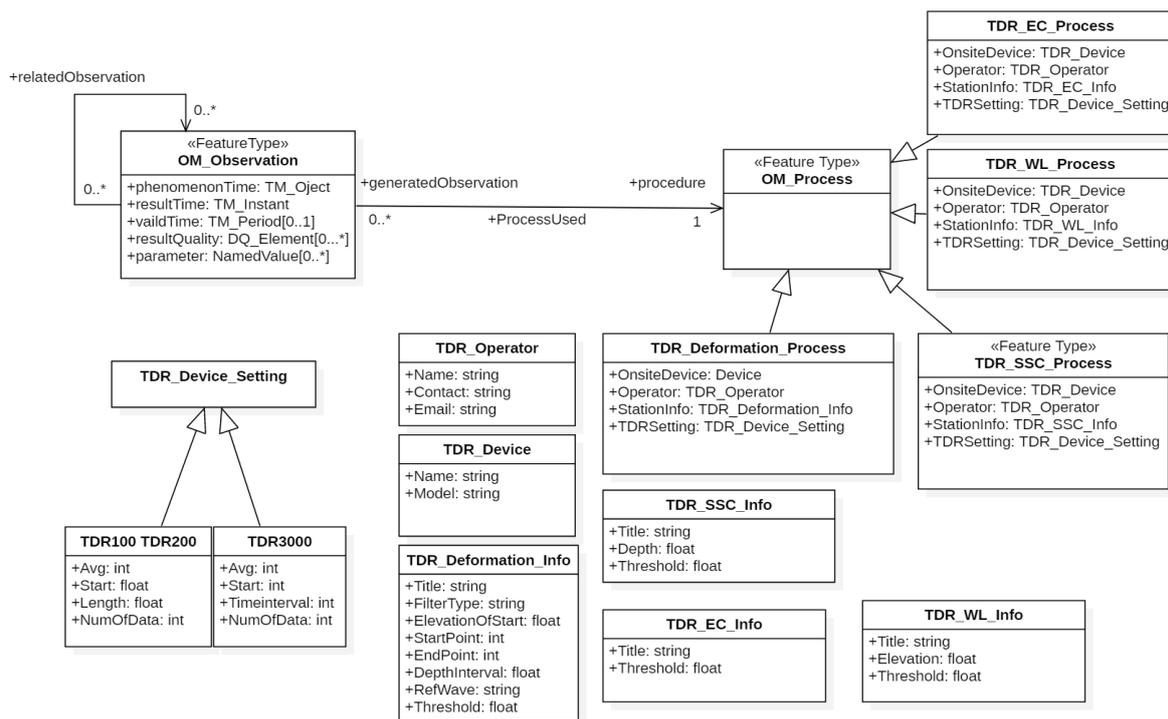


Figure 6. Class diagram of TDR OM_Process conceptual model.

Figure 7 shows the classes of TDR observations inherited from OM_ComplexObservation, which include landslide deformation observations (TDR_Deformation_Observation), suspended sediment concentration observations (TDR_SSC_Observation), electrical conductivity observations (TDR_EC_Observation), and water level observations (TDR_WL_Observation). The following describes the definitions of TDR observations:

1. TDR_Deformation_Observation: This class includes TDR measurement waveforms (Wave) in SWE DataArray format and sliding alarm data. Information of sliding warning includes sliding scale (AlarmLevel) and corresponding sliding depth (AlarmDepth).
2. TDR_SSC_Observation: This class contains the temperature of water (Temperature), the standard deviation of the temperature in water (TemperatureSTD), SSC (Concentration) and the standard deviation of the SSC (ConcentrationSTD).
3. TDR_EC_Observation: This class contains the standard deviation (ECSTD) of EC and EC itself (EC).
4. TDR_WL_Observation: This class includes water level (WaterLevel) and water level standard deviation (WaterLevelSTD).

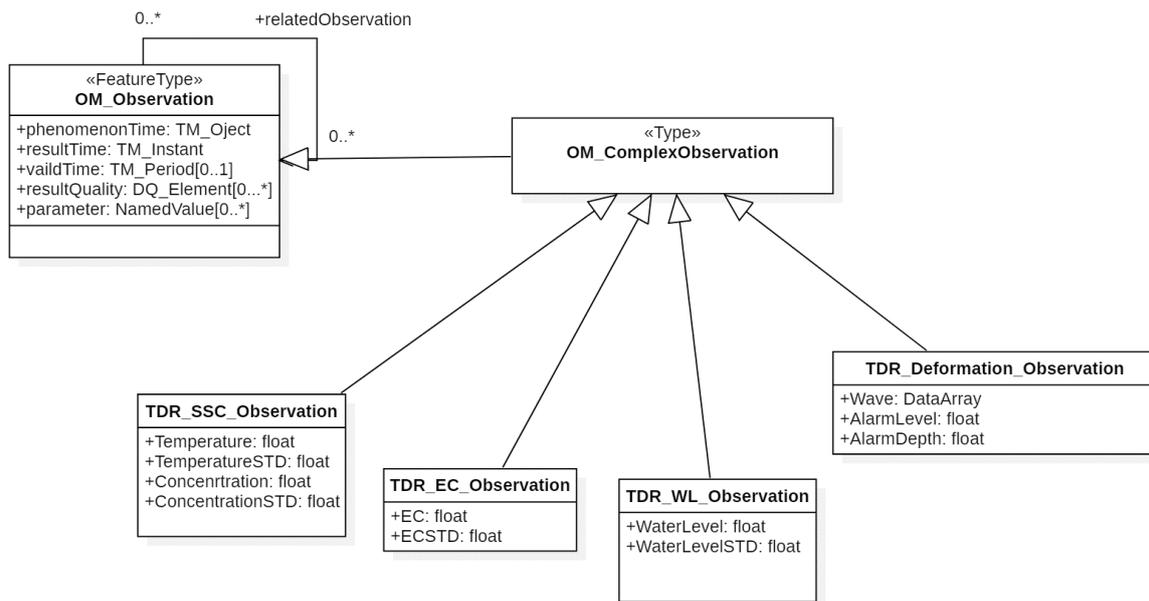


Figure 7. Class diagram of TDR OM_ComplexObservation conceptual model.

4.3. Insert Sensor

Figure 8 demonstrates the beginning of an InsertSensor request. Lines 1–5 indicate the specification version and namespaces of each referenced specification. Lines 6–7 use PhysicalSystem in SensorML v2.0 to indicate the sensor information. The identifier is to uniquely identify the sensor and this study defines the identifier as the released project name as shown in Figure 5. Lines 8–23 include the full name and abbreviation of project, where names of the research unit and project can be placed. In lines 24–33, the Offering is an important discriminating identifier for managing the stations of the project. This study defines an offeringID as the name of the monitoring station. Lines 34–39 represent the featuresOfInterest, which is a description of the observed feature. Observed features can be bridges, buildings, rivers, or air as mentioned previously. There is no restriction on this featuresOfInterest description in the SensorML 2.0 specification. This study uses the name of the monitoring station as the identifier of the observed feature, where geographic location of each monitoring station should be reported in the InsertObservation request.

Figure 9 shows the output properties. These help describe measurement parameters (ObservableProperty) of various TDR measurement types with the identifier of the monitoring station name. Each outputList contains multiple output tags. The output tag contains the name of the measurement parameter (name) and the corresponding URI definition (definition). Users can follow the URI to learn more about the TDR measurement parameters.

Regarding the format of operator information, SensorML 2.0 specification states that Contacts attribute, in the history events category, can be used to describe the manufacturer, domain expert, or anyone involved. Since the 52° North SOS implementation does not support this format, this study refers to and defines operator information in the PhysicalSystem with the parameter attribute. As shown in Figure 10, the operator information is described with the SWE DataArray format. Lines 1–10 describe the DataArray content, including basic description (swe: label and swe: description) and number of data in each record (swe: Count). Lines 11–39 define each parameter, where the definition URLs provide relevant descriptions. Lines 40–47 include the values corresponding to the defined parameters in the DataArray format.

The Physical Component category in the SensorML 2.0 specification can be used to describe any device, including overall process through the Input and Output attribute tags. However, the 52° North SOS implementation does not support this format either. This study proposes a solution similar to the idea of the operator information. A general format is shown in Figure 11.

To be specific, the TDR landslide monitoring information parameters, as shown in Figure 12, include the Name of the station, FilterType (waveform filtering method), ElevationOfStart (starting depth), StartPoint (the starting data point of the effective range of the measuring waveform, unit: positive integer), EndPoint (measure the end of the effective range of the waveform, unit: positive integer), DepthInterval (the coefficient of the depth of the waveform point conversion, positive integer unit), RefWave (reference waveform) and Threshold (the threshold value of the slope sliding, unit: reflection coefficient ranging from -1 to 1). In addition, the TDR SSC monitoring information parameters are shown in Figure 13, including Name of the station, Depth (monitoring total water depth, unit: meter) and Threshold (the threshold for mud concentration exceeding the standard, unit: parts per million).

Furthermore, the geographic location of the monitoring station is shown in Figure 14. The lines 1–8 and 9–16 indicate the longitude and the latitude of the sensor in plan view, respectively. For the case such as a directionally drilled TDR cable, the coordinates of the control points of the cable are suggested to be provided sequentially to reveal the line feature in space. Finally, the ObservableProperty and metadata description are shown in Figure 15. Lines 1–12 include the ObservableProperty measured by the TDR station. Since the SOS specification states that ObservableProperty content must provide a URI definition, this study hosts the URLs in Figure 15 to describe each ObservableProperty content. Lines 13–24 represent the format specification for InsertObservation supported by this SOS version.

```

1 <env:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.w3.org/2003/05/soap-env
2 <env:Body>
3 <swes:InsertSensor xmlns:swe="http://www.opengis.net/swe/2.0" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:gmd=
4 <swes:procedureDescriptionFormat>http://www.opengis.net/sensorml/2.0</swes:procedureDescriptionFormat>
5 <swes:procedureDescription>
6 <sml:PhysicalSystem gml:id="jiji_PubTest_Project">
7 <gml:identifier codeSpace="uniqueID">jiji_PubTest_Project</gml:identifier>
8 <sml:identification>
9 <sml:IdentifierList>
10 <sml:identifier>
11 <sml:Term definition="urn:ogc:def:identifier:OGC:1.0:longName">
12 <sml:label>longName</sml:label>
13 <sml:value>jiji_PubTest_Project</sml:value>
14 </sml:Term>
15 </sml:identifier>
16 <sml:identifier>
17 <sml:Term definition="urn:ogc:def:identifier:OGC:1.0:shortName">
18 <sml:label>Name</sml:label>
19 <sml:value>jiji_PubTest_Project</sml:value>
20 </sml:Term>
21 </sml:identifier>
22 </sml:IdentifierList>
23 </sml:identification>
24 <sml:capabilities name="offerings">
25 <sml:CapabilityList>
26 <sml:capability name="offeringID">
27 <swe:Text definition="urn:ogc:def:identifier:OGC:offeringID">
28 <swe:label>offeringID</swe:label>
29 <swe:value>jiji_PubTest_Project_offering</swe:value>
30 </swe:Text>
31 </sml:capability>
32 </sml:CapabilityList>
33 </sml:capabilities>
34 <sml:featuresOfInterest>
35 <sml:FeatureList definition="http://www.opengis.net/def/featureOfInterest/identifier">
36 <swe:label>jiji_PubTest_Project_featuresOfInterest</swe:label>
37 <sml:feature xlink:href="jiji_PubTest_Project_featuresOfInterest"/>
38 </sml:FeatureList>
39 </sml:featuresOfInterest>

```

Figure 8. Insert sensor Extensible Markup Language (XML) request, InsertSensor_identifier, offering, and featuresOfInterest.

```

1 <sml:output>
2   <sml:outputlist>
3     <sml:output name="Deformation_Wave">
4       <sml:ObservableProperty definition="http://140.115.60.148/SOS/TDR/Deformation/Wave"/>
5     </sml:output>
6     <sml:output name="Deformation_Alarm">
7       <sml:ObservableProperty definition="http://140.115.60.148/SOS/TDR/Deformation/Alarm"/>
8     </sml:output>
9     <sml:output name="SSC_Temperature">
10      <sml:ObservableProperty definition="http://140.115.60.148/SOS/TDR/SSC/Temperature"/>
11    </sml:output>
12    <sml:output name="SSC_Concentration">
13      <sml:ObservableProperty definition="http://140.115.60.148/SOS/TDR/SSC/Concentration"/>
14    </sml:output>
15    <sml:output name="EC_Sigma">
16      <sml:ObservableProperty definition="http://140.115.60.148/SOS/TDR/EC/Sigma"/>
17    </sml:output>
18    <sml:output name="WaterLevel">
19      <sml:ObservableProperty definition="http://140.115.60.148/SOS/TDR/WaterLevel"/>
20    </sml:output>
21  </sml:outputlist>
22 </sml:output>

```

Figure 9. Insert sensor XML request of InsertSensor_outputs.

```

1 <sml:ParameterList>
2   <sml:parameter name="Operator">
3     <swe:DataArray definition="http://140.115.60.148/SOS/TDR/Operator">
4       <swe:label>Operator</swe:label>
5       <swe:description>Operator Contact</swe:description>
6       <swe:elementCount>
7         <swe:Count>
8           <swe:value>4</swe:value>
9         </swe:Count>
10      </swe:elementCount>
11      <swe:elementType name="Setting">
12        <swe:DataRecord>
13          <swe:label>Operator</swe:label>
14          <swe:field name="Title">
15            <swe:Quantity definition="http://140.115.60.148/SOS/TDR/Title" axisID="Default_axisID">
16              <swe:label>Station</swe:label>
17              <swe:uom code="Text"/>
18            </swe:Quantity>
19          </swe:field>
20          <swe:field name="Operator">
21            <swe:Quantity definition="http://140.115.60.148/SOS/TDR/Operator/" axisID="Default_axisID">
22              <swe:label>Name</swe:label>
23              <swe:uom code="text"/>
24            </swe:Quantity>
25          </swe:field>
26          <swe:field name="Contact">
27            <swe:Quantity definition="http://140.115.60.148/SOS/TDR/Operator/Contact" axisID="Default_axisID">
28              <swe:label>Contact</swe:label>
29              <swe:uom code="text"/>
30            </swe:Quantity>
31          </swe:field>
32          <swe:field name="E Mail">
33            <swe:Quantity definition="http://140.115.60.148/SOS/TDR/Operator/E-Mail" axisID="Default_axisID">
34              <swe:label>E-Mail</swe:label>
35              <swe:uom code="text"/>
36            </swe:Quantity>
37          </swe:field>
38        </swe:DataRecord>
39      </swe:elementType>
40    </swe:encoding>
41    <swe:TextEncoding blockSeparator="#" tokenSeparator=","/>
42  </swe:encoding>
43  <swe:values>TestStation,Deformation,BH1,Harry,+886 924456789,harry123456@gmail.com#
44    TestStation,Deformation,BH1,Andy,+886 925123456,Andy123456@gmail.com#
45    TestStation,SSC,ch1,....#
46    TestStation,EC,ch1,....#
47    TestStation,WaterLevel,BH3,....#</swe:values>
48 </swe:DataArray>

```

Figure 10. Extended InsertSensor format for Operator.

```

1 <sml:parameter name="Device">
2   <swe:DataArray definition="http://140.115.60.148/SOS/TDR/Device">
3     <swe:label>Device</swe:label>
4     <swe:description>Onsite Device</swe:description>
5     <swe:elementCount>
6       <swe:Count>
7         <swe:value>4</swe:value>
8       </swe:Count>
9     </swe:elementCount>
10    <swe:elementType name="Setting">
11      <swe>DataRecord>
12        <swe:label>Device</swe:label>
13        <swe:field name="Title">
14          <swe:Quantity definition="http://140.115.60.148/SOS/TDR/Title" axisID="Default_axisID">
15            <swe:label>Station</swe:label>
16            <swe:uom code="Text"/>
17          </swe:Quantity>
18        </swe:field>
19        <swe:field name="Title">
20          <swe:Quantity definition="http://140.115.60.148/SOS/TDR/Title" axisID="Default_axisID">
21            <swe:label>MeasureType</swe:label>
22            <swe:uom code="Text"/>
23          </swe:Quantity>
24        </swe:field>
25        <swe:field name="Title">
26          <swe:Quantity definition="http://140.115.60.148/SOS/TDR/Title" axisID="Default_axisID">
27            <swe:label>Probetitle</swe:label>
28            <swe:uom code="Text"/>
29          </swe:Quantity>
30        </swe:field>
31        <swe:field name="DeviceName">
32          <swe:Quantity definition="http://140.115.60.148/SOS/TDR/OnsiteDevice/Device" axisID="Default_axisID">
33            <swe:label>Name</swe:label>
34            <swe:uom code="text"/>
35          </swe:Quantity>
36        </swe:field>
37        <swe:field name="Model">
38          <swe:Quantity definition="http://140.115.60.148/SOS/TDR/OnsiteDevice/Model" axisID="Default_axisID">
39            <swe:label>Contact</swe:label>
40            <swe:uom code="text"/>
41          </swe:Quantity>
42        </swe:field>
43        <swe:field name="Description">
44          <swe:Quantity definition="http://140.115.60.148/SOS/TDR/OnsiteDevice/Description" axisID="Default_axisID">
45            <swe:label>E-Mail</swe:label>
46            <swe:uom code="text"/>
47          </swe:Quantity>
48        </swe:field>
49      </swe>DataRecord>
50    </swe:elementType>
51    <swe:encoding>
52      <swe:TextEncoding blockSeparator="#" tokenSeparator=","/>
53    </swe:encoding>
54    <swe:values>TestStation,Deformation,BH1,DataLogger,CR1000,Receive measurement data and Control TDR device setting#
55      ....
56    </swe:values>
57  </swe>DataArray>

```

Figure 11. Extended InsertSensor format for InsertSensor_Device

```

1 <sml:parameter name="DeformationInfo">
2   <swe:DataArray definition="http://140.115.60.148/SOS/TDR/Deformation/DeformationInfo">
3     <swe:label>Setting parameters</swe:label>
4     <swe:description>Deformation Information</swe:description>
5     <swe:elementCount>
6       <swe:Count>
7         <swe:value>1</swe:value>
8       </swe:Count>
9     </swe:elementCount>
10    <swe:elementType name="Setting">
11      <swe:DataRecord>
12        <swe:label>Deformation Information</swe:label>
13        <swe:field name="Station">
14          <swe:Quantity definition="http://140.115.60.148/SOS/TDR/Station" axisID="Default_axisID">
15            <swe:label>Station</swe:label>
16            <swe:uom code="Text"/>
17          </swe:Quantity>
18        </swe:field>
19        <swe:field name="ProbeTitle">
20          <swe:Quantity definition="http://140.115.60.148/SOS/TDR/ProbeTitle" axisID="Default_axisID">
21            <swe:label>ProbeTitle</swe:label>
22            <swe:uom code="Text"/>
23          </swe:Quantity>
24        </swe:field>
25        <swe:field name="FilterType">
26          <swe:Quantity definition="http://140.115.60.148/SOS/TDR/Deformation/FilterType" axisID="Default_axisID">
27            <swe:label>FilterType</swe:label>
28            <swe:uom code="Text"/>
29          </swe:Quantity>
30        </swe:field>
31        <swe:field name="ElevationOfStart">
32          <swe:Quantity definition="http://140.115.60.148/SOS/TDR/Deformation/ElevationOfStart" axisID="Default_axisID">
33            <swe:label>ElevationOfStart</swe:label>
34            <swe:uom code="meter"/>
35          </swe:Quantity>
36        </swe:field>
37        <swe:field name="EffectiveRange_StartPoint">
38          <swe:Quantity definition="http://140.115.60.148/SOS/TDR/Deformation/EffectiveRange" axisID="Default_axisID">
39            <swe:label>StartPoint</swe:label>
40            <swe:uom code="Point_Number"/>
41          </swe:Quantity>
42        </swe:field>
43        <swe:field name="EffectiveRange_EndPoint">
44          <swe:Quantity definition="http://140.115.60.148/SOS/TDR/Deformation/EffectiveRange" axisID="Default_axisID">
45            <swe:label>EndPoint</swe:label>
46            <swe:uom code="Point_Number"/>
47          </swe:Quantity>
48        </swe:field>
49        <swe:field name="DepthInterval">
50          <swe:Quantity definition="http://140.115.60.148/SOS/TDR/Deformation/DepthInterval" axisID="Default_axisID">
51            <swe:label>DepthInterval</swe:label>
52            <swe:uom code="Number"/>
53          </swe:Quantity>
54        </swe:field>
55        <swe:field name="RefWave">
56          <swe:Quantity definition="http://140.115.60.148/SOS/TDR/Deformation/RefWave" axisID="Default_axisID">
57            <swe:label>RefWave</swe:label>
58            <swe:uom code="text"/>
59          </swe:Quantity>
60        </swe:field>
61        <swe:field name="Threshold">
62          <swe:Quantity definition="http://140.115.60.148/SOS/TDR/Deformation/threshold" axisID="Default_axisID">
63            <swe:label>Threshold</swe:label>
64            <swe:uom code="RefelectionCoefficient"/>
65          </swe:Quantity>
66        </swe:field>
67      </swe:DataRecord>
68    </swe:elementType>
69    <swe:encoding>
70      <swe:TextEncoding blockSeparator="#" tokenSeparator="#"/>
71    </swe:encoding>
72    <swe:values>TestStation#BH1#Moving Average#20#117#4860#0.025#0.002,0.004...,0.002#-0.3981</swe:values>
73  </swe:DataArray>
74 </sml:parameter>

```

Figure 12. Insert sensor XML request of InsertSensor_DeformationInfo.

```

1 <sml:parameter name="SSCInfo">
2   <swe:DataArray definition="http://140.115.60.148/SOS/TDR/SSC/SSCInfo">
3     <swe:label>Setting parameters</swe:label>
4     <swe:description>SSC Information</swe:description>
5     <swe:elementCount>
6       <swe:Count>
7         <swe:value>1</swe:value>
8       </swe:Count>
9     </swe:elementCount>
10    <swe:elementType name="Setting">
11      <swe:DataRecord>
12        <swe:label>SSC Information</swe:label>
13        <swe:field name="Station">
14          <swe:Quantity definition="http://140.115.60.148/SOS/TDR/Station" axisID="Default_axisID">
15            <swe:label>Station</swe:label>
16            <swe:uom code="Text"/>
17          </swe:Quantity>
18        </swe:field>
19
20        <swe:field name="ProbeTitle">
21          <swe:Quantity definition="http://140.115.60.148/SOS/TDR/ProbeTitle" axisID="Default_axisID">
22            <swe:label>ProbeTitle</swe:label>
23            <swe:uom code="Text"/>
24          </swe:Quantity>
25        </swe:field>
26        <swe:field name="Depth">
27          <swe:Quantity definition="http://140.115.60.148/SOS/TDR/SSC/Depth" axisID="Default_axisID">
28            <swe:label>Depth</swe:label>
29            <swe:uom code="m"/>
30          </swe:Quantity>
31        </swe:field>
32        <swe:field name="Threshold">
33          <swe:Quantity definition="http://140.115.60.148/SOS/TDR/SSC/threshold" axisID="Default_axisID">
34            <swe:label>Threshold</swe:label>
35            <swe:uom code="ppm"/>
36          </swe:Quantity>
37        </swe:field>
38      </swe:DataRecord>
39    </swe:elementType>
40  <swe:encoding>
41    <swe:TextEncoding blockSeparator="@" tokenSeparator=","/>
42  </swe:encoding>
43  <swe:values>TestStation,CH1,197.6,28000@</swe:values>
44 </swe:DataArray>
45 </sml:parameter>

```

Figure 13. Insert sensor XML request of InsertSensor_SSCInfo.

```

1 <sml:position>
2   <swe:Vector referenceFrame="urn:ogc:def:crs:EPSG::4326">
3     <swe:coordinate definition="SSC_CH1" name="easting">
4       <swe:Quantity axisID="x">
5         <swe:uom code="degree"/>
6         <swe:value>7.651968812254194</swe:value>
7       </swe:Quantity>
8     </swe:coordinate>
9     <swe:coordinate name="northing">
10      <swe:Quantity definition="SSC_CH1" axisID="y">
11        <swe:uom code="degree"/>
12        <swe:value>51.935101100104916</swe:value>
13      </swe:Quantity>
14    </swe:coordinate>
15  </swe:Vector>
16 </sml:position>

```

Figure 14. Insert sensor XML request of InsertSensor_position.


```

1 <sos:observation>
2   <om:OM_Observation gml:id="o2">
3     <om:type xlink:href="http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_Measurement"/>
4     <om:phenomenonTime xlink:href="#phenomenonTime"/>
5     <om:resultTime xlink:href="#phenomenonTime"/>
6     <om:procedure xlink:href="http://140.115.60.148/SOS/Project/Test"/>
7     <om:observedProperty xlink:href="http://140.115.60.148/SOS/TDR/Alarm/Level"/>
8     <om:featureOfInterest xlink:href="#Test_BH1"/>
9     <om:result xsi:type="gml:MeasureType" uom="ReflectionCoefficient">-0.22</om:result>
10  </om:OM_Observation>
11 </sos:observation>
12 <sos:observation>
13   <om:OM_Observation gml:id="o3">
14     <om:type xlink:href="http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_Measurement"/>
15     <om:phenomenonTime xlink:href="#phenomenonTime"/>
16     <om:resultTime xlink:href="#phenomenonTime"/>
17     <om:procedure xlink:href="http://140.115.60.148/SOS/Project/Test"/>
18     <om:observedProperty xlink:href="http://140.115.60.148/SOS/TDR/Alarm/Depth"/>
19     <om:featureOfInterest xlink:href="#Test_BH1"/>
20     <om:result xsi:type="gml:MeasureType" uom="Meter">34.2</om:result>
21  </om:OM_Observation>
22 </sos:observation>
23 </sos:InsertObservation>

```

Figure 17. Insert Observation XML request of TDR_Deformation_Observation (Part II).

Lines 1–16 in Figure 16 reveal the format based on the aforementioned specification, as well as the namespace for each referenced specification. Line 17 provides the offeringID that has been registered to the SOS via the InsertSensor. Lines 21–26 represent the phenomenonTime, which is the time of the measurement. As the resultTime is the time that an observation result was produced, TDR creates results at the same time that an observation was measured. Hence, the resultTime is the same as the phenomenonTime. Since TDR observations happen at individual time points, this study uses TimeInstant to describe the measurement time.

Line 27 indicates the Procedure of this observation, which was measured to the SOS via the InsertSensor. Line 28 shows ObservedProperty of this observation, which should also follow a registered ObservedProperty in the InsertSensor. Lines 29–41 represent the FeatureOfInterest of this observation, where the name tag of identifier indicates the station name, and SampledFeature describes this feature. For example, we define a project name as Test, station name as TestStation, and measurement type as Deformation. The actual geographic location is revealed in the Geography Markup Language (GML) geometry format. GML is the XML grammar defined by the OGC to express geographical features [55]. Line 42 is the actual measurement data which uses string data type to represent a TDR deformation wave observation. The data content is divided into reflected signals by using a comma (,), and arranged in order according to the measured depth. If users want to know the depth of measurement corresponding to each reflection coefficient, they can obtain the measured depths and the conversion coefficient from the DeformationInfo in InsertSensor. For example, the depth of the second point is $2 * 0.025 = 0.05$ meters where depth is (Data Point)* conversion coefficient (DepthInterval).

Figure 17 shows the continued content of this TDR displacement measurement including the alert level (Alarm_Level) and alert depth (Alarm_Depth). Each sos: Observation represents an individual observation, where the om:ObservedProperty tag defines the observed phenomenon. In addition, as the phenomenonTime, resultTime, and FeatureOfInterest have been defined previously in the TDR observation, these attributes can simply reference to the defined values. Finally, the data type of result tags is gml:MeasureType, where the uom is the unit of measurement for the result value.

5. System Implementation and TDR Data Visualization

With the proposed TDR model, the development connection scheme between TDR monitoring information and an SOS service, as depicted in Figure 18, mainly includes:

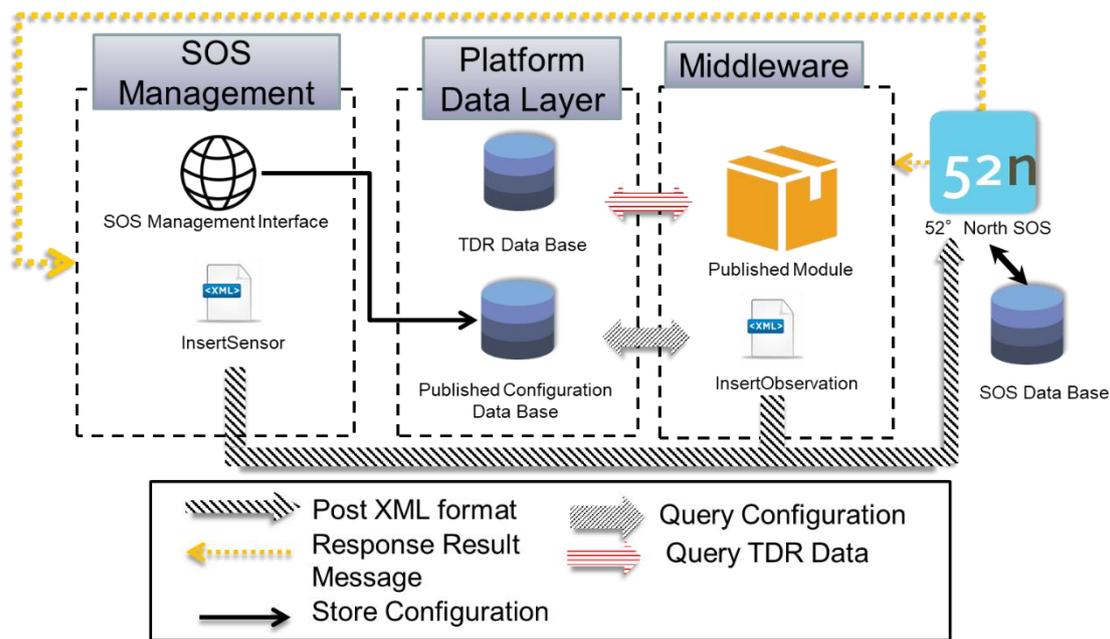


Figure 18. Connection schema between TDR information system and SOS.

1. SOS Management Interface:

SOS Management Interface is a client-side interface for web publishing and data sharing. First, the user can register sensor information through the interface settings of the 52° North SOS, and the interface stores the successful registration settings in the published configuration database.

2. SOS Middleware:

The middleware continuously publishes TDR observations to the 52° North SOS. This intermediary software will automatically query various TDR observations from TDR platform database [24] and finally publish observations to the 52° North SOS service.

3. 52° North SOS service:

This study applies the 52° North SOS implementation to support the two versions of SOS standards [31,52]. To be specific, this study mainly inserts and queries sensor descriptions and sensor observations from an SOS service.

Figure 18 depicts that the user firstly establishes the data sharing settings through the SOS Management Interface and sends registration of TDR monitoring descriptions to the 52° North SOS service through the background procedure of the SOS Management Interface. The SOS Management Interface will access the published configuration database when successfully receiving the registration message from the SOS. The SOS Middleware periodically queries the registration message and observations of the TDR sensor from the TDR database and sends observations to the SOS service via InsertObservation. Consequently, the user can retrieve all registered procedures with the GetCapabilities operation and can also retrieve detailed sensor information via Describe Sensor and obtain sensor observations by Get Observation.

Figure 19 demonstrates TDR web interface to set aforementioned effective range and reference waveform for TDR crimp type monitoring. The date of each monitoring data, which is used to set reference waveform for the time-differential method, is displayed on left side of the page. Then a user can adjust the threshold range to the TDR waveform by multiplying the standard deviation which is obtained from the differential result between measured waveform and the reference waveform. When the cursor moves to the waveform plot, the point number and the corresponding reflection coefficient are displayed simultaneously.

TDR Deformation Setting

Station TestStation ▾ Probe BH2 ▾

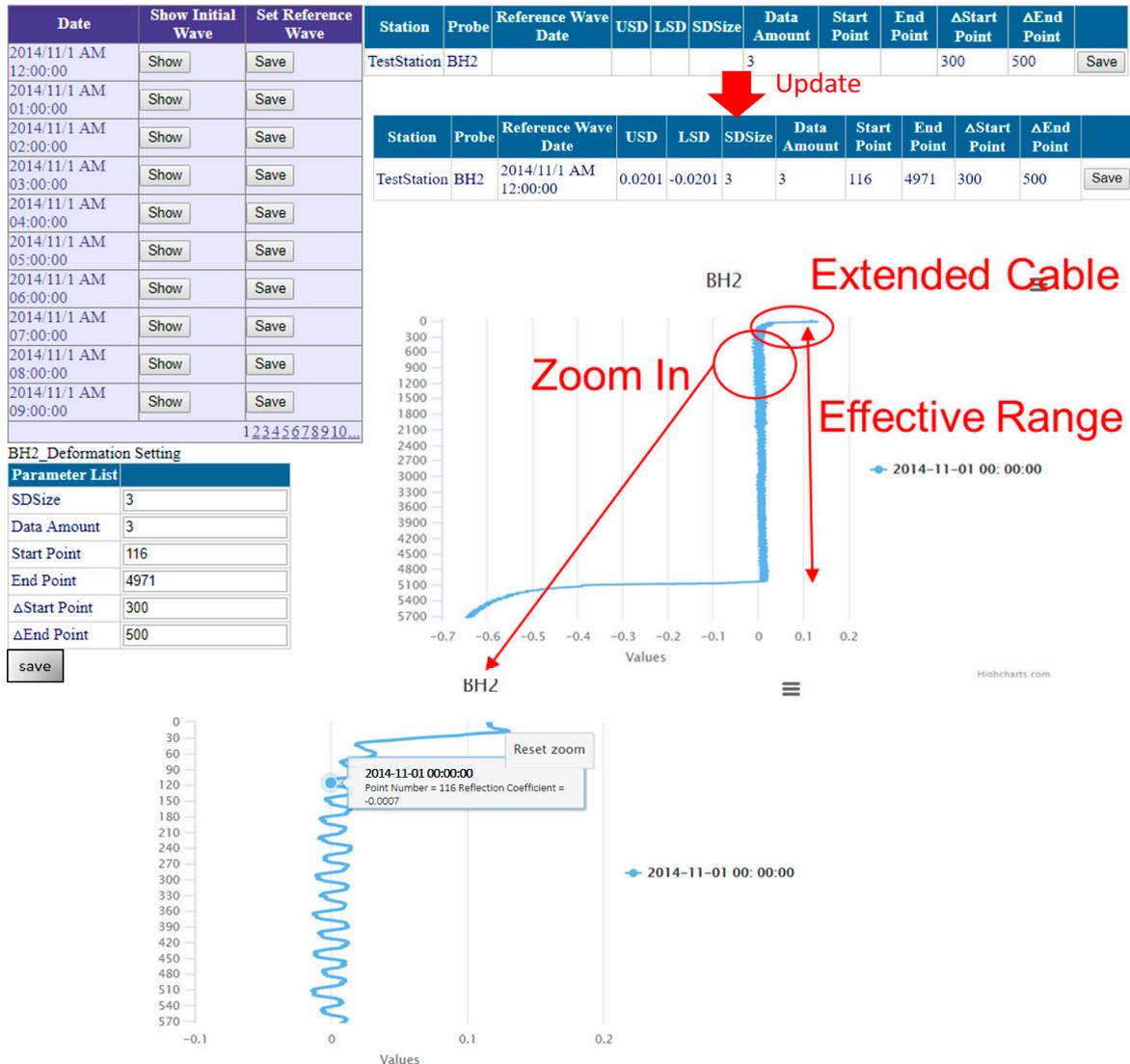


Figure 19. TDR crimp type monitoring setting for reference waveform and effective range.

Figure 20 reveals the historical data and sliding trend of TDR landslide monitoring. The part 'a' is a raw waveform data. The part 'b' demonstrates the results by the time-differential method subjected to the reference waveform, and the red line represents the criteria by setting three times standard deviation ranges of waveform noise level. The part 'c' is a sliding spike tendency according to the sliding depth. Overall, based on the proposed TDR model using SOS, all data of TDR landslide demonstrations are stored and hosted in an interoperable manner.

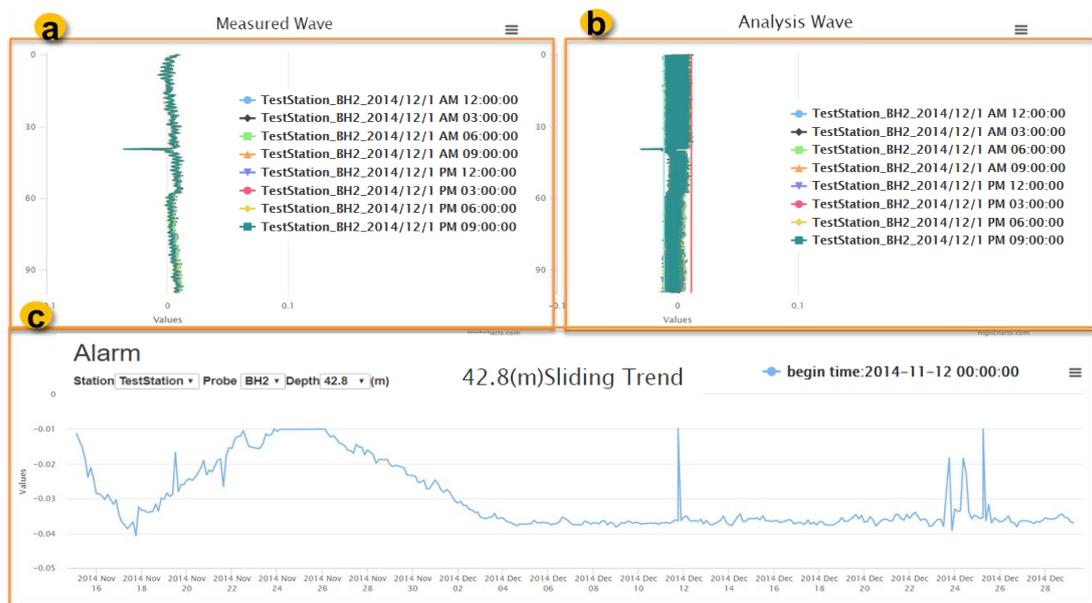


Figure 20. (a) TDR shear deformation historical raw waveform data, (b) waveform differential results, and (c) sliding trend at 42.8 m depth.

6. Conclusions

The Time-domain reflectometry monitoring system has provided a complete and diverse solution from field-side device monitoring to server-side web services. This study attempts to improve the interoperability of TDR sensor metadata and observations by proposing a TDR data model profile based on the OGC Observations and Measurements international open standard. By packaging TDR data with the well-defined O&M conceptual model, TDR data can be represented and understood unambiguously. In addition, this study implements a system prototype transforming TDR data and continuously uploading observations to an OGC Sensor Observation Service to help users retrieve TDR data in an interoperable manner. Overall, we believe that by standardizing and sharing TDR data based on the OGC Sensor Web Enablement standards, TDR data can be effectively and efficiently utilized by many important applications, such as landslide monitoring and related management based on this TDR landslide early warning system.

Author Contributions: Conceptualization, C.-C.C.; Investigation, C.-R.G.; Software, C.-R.G.; Supervision, C.-C.C. and C.-Y.H.; Validation, J.-H.J.; Visualization, C.-R.G.; Writing—original draft, C.-C.C. and J.-H.J.; Writing—review & editing, C.-Y.H.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

- O'Connor, K.M.; Dowding, C.H. *GeoMeasurement by Pulsing TDR Cables and Probes*. New York; CRC Press: Boca Raton, FL, USA, 1999.
- Robinson, D.A.; Jones, S.B.; Wraith, J.M.; Or, D.; Friedmann, S.P. A review of advances in dielectric and electrical conductivity measurement in soils using time domain reflectometry. *Vadose Zone J.* **2003**, *2*, 444–475. [[CrossRef](#)]
- Dowding, C.H.; Pierce, C.E. Use of Time Domain Reflectometry to Detect Bridge Scour and Monitor Pier Movement. In Proceedings of the Symposium on Time Domain Reflectometry in Environmental, Infrastructure, and Mining Applications, Evanston, Illinois, IL, USA, 7–9 September 1994; pp. 7–9.
- Yankielun, N.E.; Zabilansky, L. Laboratory investigation of time-domain reflectometry system for monitoring bridge scour. *J. Hydraul. Eng.* **1999**, *125*, 1279–1284. [[CrossRef](#)]

5. Yu, X.B.; Yu, X. Development and evaluation of an automation algorithm for a time-domain reflectometry bridge scour monitoring system. *Can. Geotech. J.* **2011**, *48*, 26–35. [[CrossRef](#)]
6. Wang, K.; Lin, C.-P.; Chung, C.-C. A bundled time domain reflectometry-based sensing cable for monitoring of bridge scour. *Struct. Control Health Monit.* **2019**. [[CrossRef](#)]
7. Chung, C.C.; Lin, C.P. High domain concentration suspended sediment measurements using time domain reflectometry. *J. Hydrol.* **2011**, *401*, 134–144. [[CrossRef](#)]
8. Dowding, C.H.; Huang, F.; McComb, P.S. Water pressure measurement with time domain reflectometry cables. *Geotech. Test. J.* **1996**, *19*, 58–64.
9. Thomsen, A.; Hansen, B.; Schelde, K. Application of TDR to water level measurement. *J. Hydrol.* **2000**, *236*, 252–258. [[CrossRef](#)]
10. Chung, C.C.; Lin, C.P.; Wu, I.L.; Chen, P.H.; Tsai, T.K. New TDR waveguides and data reduction method for monitoring of stream and drainage stage. *J. Hydrol.* **2013**, *505*, 346–359. [[CrossRef](#)]
11. Topp, G.C.; Davis, J.L.; Annan, A.P. Electromagnetic determination of soil water content and electrical conductivity measurement using time domain reflectometry. *Water Resour. Res.* **1980**, *16*, 574–582. [[CrossRef](#)]
12. Ledieu, J.; De Ridder, P.; De Clerck, P.; Dautrebande, S. A method for measuring soil moisture content by time domain reflectometry. *J. Hydrol.* **1986**, *88*, 319–328. [[CrossRef](#)]
13. Dirksen, C.; Dasberg, S. Improved calibration of time domain reflectometry of soil water content measurements. *Soil Sci. Soc. Am. J.* **1993**, *57*, 660–667. [[CrossRef](#)]
14. Lin, C.P.; Tang, S.H.; Chung, C.C. Development of TDR penetrometer through theoretical and laboratory investigations: 1. Measurement of soil dielectric permittivity. *Geotech. Test. J.* **2006**, *29*, 306–313.
15. Lin, C.P.; Ngui, Y.J.; Lin, C.H. Multiple analysis of TDR signal complex dielectric spectroscopy. *IEEE Trans. Instrum. Meas.* **2018**, *97*, 2649–2661. [[CrossRef](#)]
16. Giese, K.; Tiemann, R. Determination of the complex permittivity from thin-sample time domain reflectometry improved analysis of the step waveform. *Adv. Mol. Relax. Processes* **1975**, *7*, 45–59. [[CrossRef](#)]
17. Castiglione, P.; Shouse, P.J. The effect of ohmic cable losses on time domain reflectometry measurements of electrical conductivity. *Soil Sci. Soc. Am. J.* **2003**, *67*, 414–424. [[CrossRef](#)]
18. Lin, C.P.; Chung, C.C.; Huisman, J.A.; Tang, S.H. Clarification and calibration of reflection coefficient for electrical conductivity measurement by time domain reflectometry. *Soil Sci. Soc. Am. J.* **2008**, *72*, 1033–1040. [[CrossRef](#)]
19. Dowding, C.H.; Su, M.B.; O'Connor, K.M. Principles of time domain reflectometry applied to measurement of rock mass deformation. *Int. J. Rock Mech. Min. Sci. Geomech. Abstr.* **1988**, *25*, 287–297. [[CrossRef](#)]
20. Dowding, C.H.; Dussud, M.L.; Kane, W.F.; O'Connor, K.M. Monitoring deformation in rock and soil with TDR sensor cables. *Geotech. News* **2003**, *21*, 51–59.
21. Kane, W.F.; Beck, T.J.; Hughes, J.J. *Applications of Time Domain Reflectometry to Landslide and Slope Monitoring*; KANE GeoTech Inc.: Stockton, CA, USA, 2001.
22. Lin, C.P.; Tang, S.H.; Lin, W.C.; Chung, C.C. Quantification of cable deformation with TDR: Implication to landslide monitoring. *J. Geotech. Geoenviron. Eng.* **2009**, *135*, 230. [[CrossRef](#)]
23. O'Connor, K.M.; Murphy, E.W. TDR monitoring as a component of subsidence risk assessment over abandoned mines. *Int. J. Rock Mech. Min. Sci.* **1997**, *34*, 287–297. [[CrossRef](#)]
24. Hsieh, S.L.; Hung, S.H.; Chung, C.C.; Lin, C.P. Development and implementation of geo-nerve monitoring system using time domain reflectometry. *Int. Rev. Comp. Softw.* **2012**, *7*, 3238–3244.
25. Dussud, M.L. TDR Monitoring of Soil Deformation: Case Histories and Field Techniques. Master's Thesis, Northwestern University, Evanston, IL, USA, December 2002.
26. Kosnik, D.E.; Kotowsky, M.P.; Dowding, C.H.; Finno, R.J. Case studies in integrated autonomous remote monitoring. In Proceedings of the 7th International Symposium on Field Measurements in Geomechanics (FMGM), American Society of Civil Engineers, Reston, VA, USA, 24–27 September 2012.
27. Huang, C.Y.; Wu, C.H. A web service protocol realizing interoperable internet of things tasking capability. *Sensors* **2016**, *16*, 1395. [[CrossRef](#)] [[PubMed](#)]
28. Botts, M.; Percivall, G.; Reed, C.; Davidson, J. OGC[®] Sensor Web Enablement: Overview and high level architecture. In *GeoSensor Networks*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 175–190.
29. Cox, S. OGC Abstract Specification Geographic Information—Observations and Measurements 10-004r3 Version: 2.0; Open Geospatial Consortium: Wayland, MA, USA, 2013.

30. Bott, M.; Robin, A. *OGC® SensorML: Model and XML Encoding Standard 12-000 Version: 2.0.0*; Open Geospatial Consortium: Wayland, MA, USA, 2014.
31. Bröring, A.; Stasch, C.; Echterhoff, J. *OGC® Sensor Observation Service Interface Standard Implementation Specification 12-006: Version 2.0.0*; Open Geospatial Consortium: Wayland, MA, USA, 2012.
32. Taylor, P. *OGC® WaterML 2.0: Part 1—Timeseries*; Open Geospatial Consortium: Wayland, MA, USA, 2014.
33. Sofos, I.; Vescoukis, V.; Gkegkas, A.; Tsilimantou, E. Applying OGC standards to develop a land surveying measurement model. *Int. J. Geo-Inf.* **2017**, *6*, 67. [[CrossRef](#)]
34. Stasch, C.; Pross, B.; Gräler, B.; Malewski, C.; Förster, C.; Jirka, S. Coupling sensor observation services and web processing services for online geoprocessing in water dam monitoring. *Int. J. Digit. Earth* **2018**, *11*, 64–78. [[CrossRef](#)]
35. Fielding, R.T. Chapter 5: Representational State Transfer (REST). *Architectural Styles and the Design of Network-based Software Architectures*. Ph.D. Thesis, University of California, Irvine, CA, USA, 2000.
36. Liang, S.; Huang, C.Y.; Khalafbeigi, T. *OGC® SensorThings API Part 1: Sensing 15-078r6 Version 1.0.0*; Open Geospatial Consortium: Wayland, MA, USA, 2016.
37. Bray, T. *The JavaScript Object Notation (JSON) Data Interchange Format*; Internet Engineering Task Force (IETF): Fremont, CA, USA, 2017.
38. Lin, C.P. TDR as Geo-Nerve: Slope monitoring system example. *Geotech. News* **2009**, *27*, 38–40.
39. Chung, C.C.; Lin, C.P. Development of advanced TDR landslide monitoring as an early warning system validated by field implementations. *J. Eng. Geol.* **2019**. submitted.
40. Cox, S. *OGC® Observations and Measurements-XML Implementation 10-025r1: Version 2.0.0*; Open Geospatial Consortium: Wayland, MA, USA, 2011.
41. International Organization for Standardization, ISO 19156:2011, Geographic Information—Observation & Measurement. Available online: <https://www.iso.org/standard/32574.html> (accessed on 8 August 2019).
42. Taylor, P.; Cox, S.J.D. *OGC Observations and Measurements—JSON Implementation 15-100r1*; Open Geospatial Consortium: Wayland, MA, USA, 2015.
43. International Organization for Standardization, ISO 19115-1:2014, Geographic Information—Metadata—Part 1: Fundamentals. Available online: <https://www.iso.org/standard/53798.html> (accessed on 8 August 2019).
44. Gudgin, M.; Hadley, M.; Mendelsohn, N.; Moreau, J.J.; Nielsen, H.F.; Karmarkar, A.; Lafon, Y. *SOAP Version 1.2 Part 1: Messaging Framework*, 2nd ed.; W3C Recommendation: Cambridge, MA, USA, 2007.
45. Priest, M.; Na, A. *Sensor Observation Service 06-009r6 Version: 1.0*; Open Geospatial Consortium: Wayland, MA, USA, 2007.
46. Song, X.; Wang, C.; Kagawa, K.; Raghavan, V. Real-time monitoring portal for urban environment using sensor web technology. In *Proceedings of the 18th International Conference on Geoinformatics, Beijing, China, 18–20 June 2010*; pp. 1–5.
47. Sorg, J.; Kunkel, R. Conception and implementation of an OGC-compliant sensor observation service for a standardized access to raster data. *Int. J. Geo-Inf.* **2015**, *4*, 1076–1096. [[CrossRef](#)]
48. Whiteside, A. *OGC Web Services Common Specification 06-121r3*; Open Geospatial Consortium: Wayland, MA, USA, 2007.
49. Echterhoff, J. *OpenGIS® SWE Service Model Implementation Standard 09-001 Version: 2.0*; Open Geospatial Consortium: Wayland, MA, USA, 2011.
50. International Organization for Standardization, ISO 19143:2010, Geographic information—Filter Encoding. Available online: <https://www.iso.org/standard/42137.html> (accessed on 8 August 2019).
51. Pearlman, J.; Simon, J.; Rio, J.; Delory, E.; Frommhold, L.; Martinez, S.; O’Reilly, T. Oceans of tomorrow sensor interoperability for in-situ ocean monitoring. In *Proceedings of the Oceans 2016 MTS/IEEE Monterey, Monterey, CA, USA, 19–23 September 2016*; pp. 1–8.
52. North, Standardized, Web-Based Upload and Download of Sensor Data and Sensor Metadata. Available online: <https://52north.org/software/software-projects/sos/> (accessed on 8 August 2019).
53. Open Geospatial Consortium. *MapServer 7.4.1 Documentation, OGC Support and Configuration*. Available online: http://mapserver.org/ogc/sos_server.html (accessed on 8 August 2019).

54. Deegree, Deegree SOS Online. Available online: <http://deegree.org/> (accessed on 15 December 2017).
55. Portele, C. OGC® *Geography Markup Language (GML)—Extended Schemas and Encoding Rules*; Open Geospatial Consortium: Wayland, MA, USA, 2012.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).