Computers & Geosciences 56 (2013) 45–55

Contents lists available at SciVerse ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

A direct registry service method for sensors and algorithms based on the process model

Nengcheng Chen *, Xiaolei Wang, Xunliang Yang

State Key Lab for Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, 129 Luoyu Road, Wuhan 430079, China

ARTICLE INFO

Article history:
Received 3 December 2012
Received in revised form
5 March 2013
Accepted 7 March 2013
Available online 21 March 2013

Keywords:
Sensor registry
Algorithm registry
Catalogue information model
Process model
CSW

ABSTRACT

To characterize and monitor environmental quality in real time, significant efforts must be made to make sensors accessible on the Web. Although a number of services have shown great potential to improve the accuracy and reliability of sensor registry, comprehensive methods that register sensors and algorithms in a catalogue service based on a process model are lacking. This paper addresses the registry of processes, including physical and non-physical processes, by using the OGC catalogue service. To provide more information on sensors and algorithms, the process catalogue information model extends the ebXML Registry information Model. The mapping between the metadata encoded in SensorML and the extended model includes more information for all types of processes. Moreover, the mapping includes the relationship among processes and their categories based on the properties of sensors and algorithms. The effectiveness of the improved model is registered and implemented in GEOSENSOR CSW, following the OGC "transaction" interface. Flooding experimental results demonstrate that the service can effectively and directly register sensors and algorithms so that processes can be flexibly executed according to users' demands.

1. Introduction

Existing sensors, including in-situ, airborne, and spaceborne sensors, will be Web-ready in the future. A large number of algorithms exist to describe the mathematical operations of sensors. Sensors and algorithms require effective management on the Internet. In the Sensor Web, the metadata for data and services could be registered in a catalogue that enables users to locate, access, discover, and make use of such resources through the catalogue service (Chen et al., 2009; Pandey and Patel, 2009). These methods (Park et al., 2007; Parhi et al., 2010) provide facilities for retrieving, storing, and managing numerous kinds of resource. The management of sensors and algorithms based on the process model remains lacking, and will become an increasingly urgent matter in the Earth observation community. Therefore, a catalogue service must be built to manage processes given the large number of sensors and algorithms in use.

Processes in Sensor Web Enablement derived from AbstractProcess are conceptually divided into two types (Botts and Robin, 2007; Botts et al., 2007; Chen et al., 2012): (1) non-physical or "pure" processes (ProcessModel and ProcessChain), which can be treated merely as mathematical operations and (2) physical processes (Component and System), such as detectors, actuators, and sensor systems, where information regarding their positions and interfaces may be relevant. Processes are encoded in standard models and in XML Schema, which is the core information model (SensorML) in the Sensor Web. SensorML provides descriptions of sensors and algorithms needed for sensor management as well as for the location of sensor observations, processing of low-level sensor observations, and listing of testable properties (Broring et al., 2011). All processes in SensorML can provide inputs, outputs, parameters, and additional information. The information required for resource discovery, qualification of results, and for assistance to humans are included within MetadataGroup, for example, the properties of identification, classification, and capabilities (Jirka et al., 2009). So the registry of process models should include the above information.

The proposed registry service has a specific registry model which needs a normative XML Schema for the representation of registry objects. Current registry models include Universal Description, Discovery, and Integration (UDDI) (Clement et al., 2004) and the ebXML registry information model (ebRIM), both of which are sponsored by the Organization for the Advancement of Structured Information Standards (OASIS). The UDDI specification does not adequately facilitate the autonomous discovery and interoperation of disparate Web services, while ebRIM enables the sharing of content and metadata among organizational entities in a federated environment (Fuger et al., 2005). The ebRIM constitutes a public schema that specifies how the catalogue content should be structured and interrelated for discovery and publication purposes, using a set of classes and the relationships

* Corresponding author. Tel.: +86 13886019231; fax: +86 27 68778229.
E-mail address: cnc@whu.edu.cn (N. Chen).

0098-3004/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cageo.2013.03.008

among these classes (Chen et al., 2009).The process registry service would use ebRIM to specify the metadata for process models. But it is necessary to provide a common vocabulary that extends the abstract information model (ebRIM) and supports the display, retrieval, and search of the processes in process registry service. The CSW-ebRIM profile, which is developed by the Open Geospatial Consortium (OGC), provides guidance for the registration of metadata. The proposed registry service could apply the OGC–CSW interfaces to ebRIM to provide a general and flexible Web-based registry service (Martell, 2009).

Some existing catalogues currently store metadata describing resources and allow users to find these resources (Pandey and Patel, 2009; Park et al., 2007; Parhi et al., 2010). Houbie and Bigagli (2010) proposed service interfaces and encodings of catalogue service for ebRIM application profile for managing Earth Observation data products. The method done by Houbie and Begali is easy to use and expand, while the registry objects only address the data products for earth observation. Deegree-CSW (Deegree, 2010, 2011) is a simple but efficient test tool for CSW transaction requests and responses with high performance and excellent scalability. However, the service lacks CSW harvest support and could be slow because of the heavy use of XSLT transformations over procedural code (Deegree, 2010). GeoNetwork CSW (GeoNetwork, 2010) is used for publishing and accessing digital catalogues of metadata for geospatial data, services, and applications. This service provides an easy-to-use Web interface for the search of geospatial data and connects distributed map services for publishing and discovery. Although the core queryable properties in the GetRecords response include spatial reference elements, the spatial filter is only applied to LatLongBoundingBox and not to other formats for spatial relationships. Researchers at the Laboratory for Advanced Information Technologies and Standards at George Mason University (GMU, 2006) are also building a catalogue service based on OGC standards. The catalogue service enables the Earth-science community to exchange geospatial resources by searching for pre-registered spatial and temporal metadata information that can be used for the publication and discovery of geospatial services and data (GMU, 2010); (Yue et al., 2011). However, the tool only contains the basic operations of GetCapabilities, GetRecords, and GetRecordsByID, which fall short of users' requirements. The current 52North Sensor Instance Registry release (Jirka and Nust, 2010), enables users to insert metadata, search for sensors, handle the status of sensors, and link SIR instances to OGC Catalogs (52 North, 2012). It can manage the metadata as well as the status information of sensors in a manner compatible with OGC catalogs. However, the administrators of SIR must establish a connection with a generic catalogue service, such as CSW, to enable the regular publishing of all data to a certain catalogue instance. These systems have several limitations.

(1) Some sensor registry systems must establish a connection between the sensor catalogue and a generic catalogue service (such as the OGC–CSW). When the system establishes a link with CSW through the catalogueID, clients can discover sensors which are registered by data providers into the sensor catalogue. The clients cannot register the sensor and obtain the query results directly through CSW.
(2) Current sensor registry services do not include enough information to satisfy the users' requirements with a unified and standardized model for storing and managing heterogeneous sensors. For example, SIR only defines the mappings between SensorML and ebRIM that contains *System* and *Component* objects. The mappings include the definitions of *ID*, *long Name*, *short Name*, *description*, *observedBBOX*, *input*, *output*, *location*, *validtime*, lacking of *parameters* and *connections*. When users require the accurate discovery of a specific sensor with given parameters, the catalogue service cannot return the corresponding results.
(3) Currently implemented CSW systems cannot register and discover all types of processes. These services (for example, the work done by Houbie et. al. in OGC 09-163r2) only provide a physical process registry and do not support the registration of all processes, especially *ProcessModel* and *ProcessChain*.

This paper addresses the aforementioned issues. All process types could be registered directly in an on-demand catalogue service, which contains more information on sensors compared with the existing implementations such as the 52N-SIR. The mappings in this paper demonstrate the management of algorithms in the form of process model. The remainder of this paper is organized as follows: Section 2 introduces the method used to design the process catalogue schema for the representation of the registry content and the main mappings from SensorML to the processes. The implementation of the registry service and registry experiments are presented in Section 3. In Section 4, the characteristics of the proposed registry method are discussed. Section 5 summarizes the findings and discusses the next steps.

2. Enabling process registration

2.1. Catalogue information model

The *rim*:*RegistryObject* is the core metadata class and serves as a common super class for most classes in ebRIM. A *RegistryPackage* instance is a logical collection of related *rim*:*RegistryObject* instances as its members (Wilson, 2007). The *rim*:*RegistryPackage* is a formal method of representing an extension package through a set of elements and extensibility points that enables it to be tailored for specific purposes (Houbie and Bigagli, 2010; Houbie et al., 2010). The extension package provides several standard classification schemes such as *ObjectType*, *AssociationType*, *ClassificationSchemeType* and *SlotType*. These extensibility points, which include new extrinsic Objects, associations, classifications, and slots, could be used to represent the Registry Object metadata for information resources (Fuger et al., 2005).

According to that, the *Process_Catalogue* is created as *rim*:*RegsitryPackage*, which is not explicitly implemented in the process registry. *Process_Catalogue* package members are RegistryObjects which include the extensibility points. In the paper, the creation of extensions store information on processes in the catalogue for various sensors and algorithms. And the extension points include the core attributes that are useful for discovering processes encoded in SensorML, as mentioned below.

An overview of the catalogue information model of the process registry based on ebRIM 3.0 is shown in Fig. 1. Note that in Fig. 1 the classes by white icons come from the ebRIM 3.0; the classes by yellow icons are the same as the classes in OGC 09-163r2 (Houbie et al., 2010); the classes in green icons have additions compared to the classes defined in OGC 09-163r2; the new classes by red icons are created in this paper.

2.2. The extension based on ebRIM for process model

2.2.1. Additional ExtrinsicObject types

AbstractProcess could be divided into two types: physical processes and non-physical (pure) processes, as mentioned in Section 1. These classes (*AbstractProcess*, *PhysicalProcess* and *NonPhysicalProcess*), are conceptual and are mapped from abstract concepts in [OGC 07-000], so they do not appear as explicit classes when implemented. Accordingly, three abstract classes(*AbstractProcess*, *AbstractPhysicalProcess* and Abstract*NonPhysicalProcess*) are created, as shown in Fig. 1.