SENSOR DATA PUBLICATION ON THE WEB FOR SCIENTIFIC APPLICATIONS

Gilberto Zonta Pastorello Jr., Luiz Gomes Jr., Claudia Bauzer Medeiros
Institute of Computing, UNICAMP, Av Albert Einstein, 1251 Campinas, SP, Brazil

André Santanchè
DCEC – UNIFACS, Av Cardeal da Silva, 747 Salvador, BA, Brazil

Keywords: Data publication on the Web, Web services standards, Scientific data, Sensor data.

Abstract: This paper considers the problems of sensor data publication, taking advantage of research on components and Web service standards. Sensor data is widely used in scientific experiments – e.g., for model validation, environment monitoring, and calibrating running applications. Heterogeneity in sensing devices hamper effective use of their data, requiring new solutions for publication mechanisms. Our solution is based on applying a specific component technology, *Digital Content Component* (DCC), which is capable of uniformly encapsulating data and software. Sensor data publication is tackled by extending DCCs to comply with geospatial standards for Web services from OGC (*Open Geospatial Consortium*). Using this approach, Web services can be implemented by DCCs, with publication of sensor data following standards. Furthermore, this solution allows client applications to request the execution of pre-processing functions before data is published. The approach enables scientists to share, find, process and access geospatial sensor data in a flexible and homogeneous manner.

1 INTRODUCTION

Sensors are fast becoming one of the main data providers for scientific applications. A given set of sensors may provide data to meet the needs of distinct applications – e.g., rainfall and temperature sensors may be used by researchers in environmental planning, habitat monitoring or epidemiology. However, each application domain – and each application within a domain – will require distinct kinds of data granularity and sampling. So, the question we answer is the following: how to devise a solution to the problem of sensor data publication, to support homogeneous access mechanisms to geospatial sensor based data from heterogeneous sources.

Our solution is based on a specific component technology, *Digital Content Components* (DCCs), which we extended to comply with geospatial Web service standards from OGC (*Open Geospatial Consortium*). DCCs are capable of uniformly encapsulating data and software, and are annotated with metadata and references to ontologies, following Semantic Web standards. We use them to encapsulate access to sensing data, homogeneously integrating them into a single framework. DCCs provide basic data manipulation functions, and can be composed into arbitrarily complex procedures. We use these functions to propose an extension to the OGC standards and proceed to show how to use DCCs to implement these extended standards.

We use a running example based on epidemics monitoring of dengue fever. It is caused by a virus and is transmitted to humans by the *Aedes aegypti* mosquito. Efforts to monitor dengue epidemics require combining geospatial data such as registered disease cases, geographical and demographic characteristics of the population, environmental data that is known to affect disease spread (e.g., rainfall or temperature), and locations where the mosquito is found.

The acquisition and access to such environmental readings for experimental research is an open issue. Our work contributes towards solving these problems.

The rest of the paper is organized as follows. Section 2 presents the basics of the DCC technology and explains the encapsulation of resources into DCCs. Section 3 describes our proposal for multi-level in-

Zonta Pastorello Jr. G., Gomes Jr. L., Bauzer Medeiros C. and Santanchè A. (2008).
SENSOR DATA PUBLICATION ON THE WEB FOR SCIENTIFIC APPLICATIONS.
In *Proceedings of the Fourth International Conference on Web Information Systems and Technologies*, pages 137-142
DOI: 10.5220/0001515301370142

tegration of sensor data Web publication. Section 4 discusses implementation issues. Section 5 considers related efforts. Section 6 presents concluding remarks and ongoing work.

2 DCCS AND RESOURCE ENCAPSULATION

This section briefly presents DCCs (section 2.1) and explains how we apply them to homogeneously encapsulate three kinds of resources: data, data sources (e.g., databases and sensors), and software.

2.1 DCC Basics

A *Digital Content Component* (DCC) is a unit of content and/or process reuse, which can be employed to design complex digital artifacts. It can be seen as digital content (data or software) encapsulated into a semantic description structure. It is comprised of four sections (Figure 1):

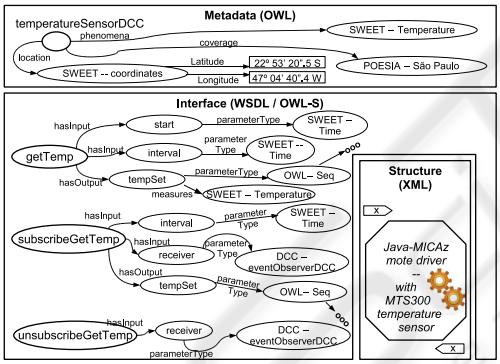

Figure 1: A DCC for sensor access.

(1) the content itself (data or code, or another DCC), in its original format. In the example, the content is a driver for communicating and gathering data from a MICAz[1] sensor;
(2) the declaration, in XML, of a structure that defines how DCC internal elements relate to each other (here, delimitating the object code of the sensor's driver);
(3) specification of an interface, using adapted versions of WSDL and OWL-S – e.g., `getTemp` and `subscribeGetTemp` operations;
(4) metadata to describe functionality, applicability, etc., using OWL (the DCC is declared as belonging to class `TemperatureSensorDCC` and located at `longitude` and `latitude` specified).

[1] www.xbow.com/Products/productdetails.aspx?sid=101

Interface and metadata are linked to ontology terms – e.g., input parameters of the `getTemp` operation are timestamps defined by the "Time" concept of NASA's SWEET (Raskin and Pan, 2003) ontology.

There are two main kinds of DCC – process and passive. The first encapsulates any kind of process description, and Passive DCCs consist of any other kind of content (e.g., a text or video file). See (Santanchè et al., 2007) for details.

2.2 Encapsulation of Data (Access and Manipulation)

We encapsulate data withing PassiveDCCs and sensing sources in ProcessDCCs. DCCs can be used to homogeneously publish any kind of sensor generated data, be it static and/or streamed data (Pastorello Jr et al., 2007) . A satellite image, or a file containing a temporal series of temperature data, are typical examples of static data, while continuous temperature reading transmissions are an example of streamed data. These DCCs are annotated using ontology terms such as data type, the physical phenomenon being measured (temperature, solar radiation, etc), the geographical location of the reading (e.g., GPS-provided). See (Pastorello Jr et al., 2007) for more details on data encapsulation.

The number of sensors encapsulated within a ProcessDCC depends only upon the implementation of the sensor driver (as the MICAz driver in Figure 1). A sensor network, for instance, can be encapsulated by a DCC with a driver that communicates with the network's access point.

Manipulation of sensor data can occur in two levels: within a sensor or a network (signal processing, in-network fusion, etc) or externally (filtering data by region, fusion of heterogeneous water temperature sensors, etc). External processing, sometimes called post-processing, is usually application oriented – e.g., summarization of temperature readings per region and time period for a dengue spread simulation.

3 SENSOR DATA PUBLICATION

In most cases, sensor data is georeferenced. This makes it possible to employ general-use geospatial standards and services for sensor data publication (section 3.1). Section 3.2 shows how to use DCCs to publish sensor data in different scenarios. DCC annotations are translated into Web standards-compliant metadata, and DCCs are used to implement Web services (section 3.3).

3.1 OGC Standards

3.1.1 Geospatial Data Publication

The *Open Geospatial Consortium* (OGC) is an international organization that leads the development of standards for interoperability among geospatial applications. OGC's main general-use standards for geospatial data interoperability are the *Web Feature Service* (WFS), the *Web Coverage Service* (WCS) and the *Web Map Service* (WMS) (OGC, 2007a). A central notion is that of *feature*, i.e., a geospatial object. These standards specify the access mechanisms to, respectively, vector data (point-line-polygon), raster data (image-based) and rendered maps. These standards are specified to be implemented as Web services.

The WFS specification provides a standardized means to access geospatial data encoded in GML (*Geographic Markup Language*) (OGC, 2007a) for the transport and storage of georeferenced data. A WFS-compliant service implements operations that allow retrieval of data and metadata, using several kinds of filters. The WCS specification allows interactions similar to these of WFS, but for raster data. Finally, the WMS specification allows clients to pose queries to retrieve rendered maps. Queries can specify a map's geographic extent, output format and the style – which is defined as *Style Layer Descriptor* (SLD) (OGC, 2007a) files.

Roughly speaking, a query to retrieve Features (WFS), Coverages (WCS) or Maps (WMS) can be expressed by a tuple <`query,filter,style` >. The `query` is subject to `filters` , and `style` is the SLD specification for maps. Section 3.2 describes our extension to this approach.

3.1.2 Sensor Data Publication

OGC is now working on standards for sensor interoperability and sensor data access and publication. Its *Sensor Web Enablement Working Group* (SWE – http://www.opengeospatial.org/projects/groups/sensorweb is proposing standards for data encoding and common Web service interfaces for data access. The encoding proposals are *Observation & Measurements Schema* (O&M), *Sensor Model Language* (SensorML or SML), and *Transducer Markup Language* (TransducerML or TML) (OGC, 2007b). As most of OGC's standards, the languages are defined by means of XML Schemas. The service interface proposals are *Sensor Observation Services* (SOS), *Sensor Planning Service* (SPS), *Sensor Alert Service* (SAS), and, *Web Notification Services* (WNS) (OGC, 2007b). The SWE Common (OGC, 2007b) initiative aims at a common vocabulary to be used within the SWE framework.

The most basic encoding is TML, which deals directly with transducer (sensor or actuator) data. Higher level encoding is covered by SML, which can represent the processes sensor data went through. The last encoding level is O&M, which represents sensor originated data independently from the level of data processing.

SML is used for modeling and representing processes that generate sensor data. SML data sources are not restricted to sensors alone, and can also be a sensor network, a sensor wrapper or database with sensor data, etc. In SML a *ProcessModel* is an atomic processing block that defines its own inputs, outputs and parameters. It is also related to a *ProcessMethod*, which defines the interface and behavior for a process as well as metadata about the data it can provide. A *ProcessChain* is a composite processing block, built upon ProcessModels or other ProcessChains.

From the service interfaces point of view, SOS is intended to provide access to sensor data represented in any of the three encoding proposals (O&M, SML and TML). SPS focus on providing access to data acquisition and manipulation capabilities from resources (e.g., processing systems, archiving systems, sensors and/or auxiliary systems). SAS and WNS are intended to provide means of subscribing to a service (SAS) for update notifications (WNS).

3.2 Accessing Sensor Data

Many interoperability issues can be solved by publication of sensor data using the analyzed OGC standards. Sensor data can be accessed by using WFS (data as a feature) or SOS (with specific mechanisms for sensor data access). In either case, access is carried out by posting a query to a standard-compliant Web service. The query and the result format standards provide means to uniformly describe, publish and access data produced by sensor devices. If WFS is used, an application domain schema must be previously agreed upon by the participants.

However, in a research scenario using sensor data, access via query posting does not always suffice. Two unresolved issues are the following:

- Scientists need to be able to request data pre-processing before executing a query;
- Good pre-processing functions used in models need to be made available to other scientists, for reuse and validation of each other's work;

OGC is trying to solve these issues by enhancing query filter mechanisms. Nonetheless, more flexible

support is also desirable. Besides the filtering option offered by OGC, we propose two novel solutions: (a) the producer should publish a list of pre-processing functions to be chosen by the consumer, or (b) the producer should allow the consumer to post the entire processing operation within a request. The latter can be achieved by posting a procedure schematics (such as a workflow).

Option (a) is easy to implement. The publication interface can provide these functions as different service operations. Nevertheless, this requires modifying the service every time a new function is to be made available. Option (b) is particularly interesting, given available standard ways to represent Web service compositions (viz., BPEL – www.oasis-open.org/committees/wsbpel) – e.g., specifying a given sequence of operations on temperature readings before publishing the data.

Therefore, we propose to extend OGC's combination <query,filter,style> to <query,schematics,style> for data access. Section 3.3 details this solution.

3.3 DCCs and the Standards

Our publication proposal adopts DCCs and OGC standards in complementary roles. On the one hand, OGC has a well established XML-based standard to represent geospatial metadata. On the other hand, DCCs adopt OWL, which opens plenty of integration possibilities.

We start by uniformly encapsulating sensor data within DCCs, which have associated annotations. These annotations can be used translating data within the DCC to any of the OGC sensor data encoding standards (detailed below).

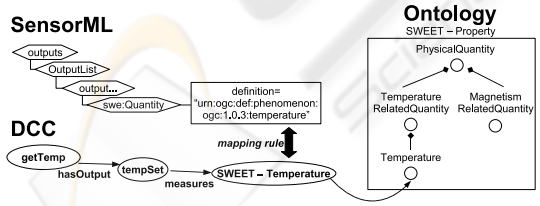

Figure 2: Mapping SML to DCC Metadata.

Consider, in our dengue example, a temperature sensor annotated with SML metadata, handled by the DCC presented in Figure 1. SML metadata can be mapped to DCC metadata by establishing a correspondence between key SML elements and ontology concepts in OWL. An example is illustrated in Figure 2, which maps a SML sensor output to a DCC output. On the upper left of the figure, the SML XML annotation defines that the sensor outputs temperature readings urn:ogc:def:phenomenon:ogc:1.0.3:temperature. On the lower left of the figure, this output is mapped to a DCC operation output: OGC XML temperature is mapped to the "temperature" concept, part of SWEET ontology (right of the figure).

Published sensor data can be combined with other kinds of data. Consider, now, combining sensor data on rainfall and temperature with data on mosquitoes and diseases. Through DCCs we create bridges between OGC temperature concepts and other ontologies for mosquitoes and diseases. Our TemperatureDCC (Figure 2) can be composed with other DCCs – e.g., its OWL metadata can be further connected with OWL ontologies for insects (mosquito) and diseases (Dengue).

Sensor data may need to be published at distinct granularities. A user may need a high level summary of mosquito locations, and also a detailed view on the rainfall data. This issue is considered by efforts in SWE. Using SWE alone, however, is not enough. SWE considers only process representation and not access to process execution and customization. With our extension, these processes become available for invocation through DCC interfaces.

DCC interfaces are described in WSDL and OWL-S and thus are compatible with Web service interfaces. This way, we achieve adherence to OGC's standards for data representation and data access. Moreover, since DCCs were originally conceived as reuse units, they support flexibility in adding new pre-processing functions. Finally, DCCs can offer access not only via SOAP messages (in a Web environment), but also be used in a standard programming environment.

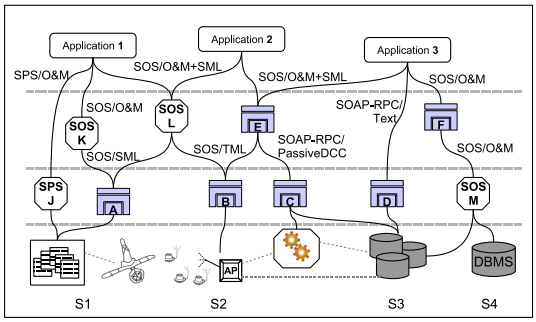

Figure 3: A multi-level integration example scenario combining DCC and SWE.

Figure 3 shows an example scenario combining data access using (i) specific implementations of OGC's publication standards (octahedrons) – e.g., SOS/SML, SPS/O&M; and (ii) DCCs (squares),

using DCC communication mechanisms – e.g., RPC/PassiveDCC – and OGC publication standards – e.g., SOS/O&M, for Web service implementations. The labels in the lines show which communication mechanism and which sensor data encapsulation strategy are employed. Each level passes data through a processing function – from raw production to complex manipulations. For instance, specific pre-processing can be invoked to filter out outliers, summarize data or merge data from several sources. Basically, one can design distinct DCCs dedicated to each such pre-processing step, and application designers can dynamically choose which implementation to adopt.

Applications and other processing services may access data available in any level. Consider *Application 3*, for instance. Let `S1` through `S4` be data sources, where `S1` are satellite images from which vegetation can be derived, `S2` is a network of temperature sensors, `S3` is a database of current infection case data, and `S4` is historical infection data. The application corresponds to a scenario of map generation using: sensor location details (from **B** and **E**), sensor fused temperature data (from **C** and **E**), infection case data (from **D**), and time-series data on monthly distribution of past cases (from **M** and **F**). Maps generated by *Application 3* can be a final result – the spatial distribution simulating new case occurrences for the next month. They also can be used to feed another application.

4 IMPLEMENTATION

A few components were implemented to illustrate the main ideas. Two sensor platforms, TelosB[2] and MICAz, and a database system, PostgreSQL, were used as data sources. One sensor data processing unit was implemented, using a simple summarization algorithm. The GeoServer http://www.geoserver.org/) implementation of the Feature-Map-Coverage (FMC) services, i.e., a WFS/WMS/WCS server, was used to publish data as a service. The MapBuilder http://communitymapbuilder.osgeo.org/ tool was combined with the DOJO Javascript toolkit http://dojotoolkit.org/ to enable Web browser visualization.

The code running on the sensors was implemented in NesC (Gay et al., 2003), a C-derived component oriented programming language, using the TinyOS (Levis et al., 2004) interfaces. Data access on the sensors was carried out by ProcessDCCs with encapsulated drivers, implemented in Java.

A simple application was developed using these components. Temperature data is collected by the sensors and acquired by the respective ProcessDCCs. Operations of these DCCs allow access to the data in real-time. Other ProcessDCCs access these data for making them available as raw real-time data, summarized data, and time-series (stored on the database system). The stored time-series are available to the FMC server through the database and real-time data are available directly.

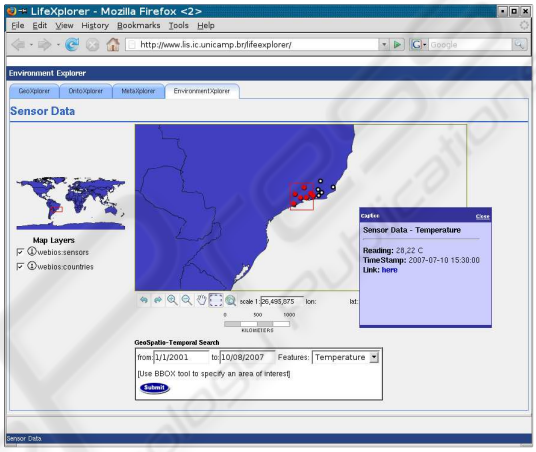

Figure 4: A screen capture of the application.

Figure 4 shows a screen capture of the client application. Air temperature data is available in real-time and as time-series, both with the option of using the summarization feature.

5 RELATED EFFORTS

Efforts related to our work include scientific and sensor data manipulation and publication.

General approaches are also possible. The ones considered are: (i) Specialized implementations; (ii) Software components and communication middlewares, such as CORBA, COM+ and .NET, EJB, and others. The first approach has the classic overhead of unnecessary repetition of work, hard maintenance, poor standardization and interoperability. Components and middleware lack flexibility, semantic descriptions, and, more importantly, homogeneous treatment of data, sources and software.

(Iamnitchi et al., 2002) address collaboration in a peer-to-peer scientific data sharing scenario. They claim that emerging patterns typical of scientific collaboration can be exploited to improve data sharing and search mechanisms. Although we are not

[2] www.xbow.com/Products/productdetails.aspx?sid=252

concerned with network organization issues, DCC's caching mechanisms (Santanchè et al., 2007) have a similar effect, reducing latency time for frequently accessed resources.

(Aloisio et al., 2006) propose a grid architecture to integrate sensor networks. The architecture regards sensors as grid resources and employs SML to describe them. The proposal does not consider publication of sensor data outside their grid infrastructure, hampering data reuse. (Chu et al., 2006) follow a similar approach, employing grid technology. Their architecture is strongly based on OGC standards for sensor data, which is in line with our approach.

6 CONCLUDING REMARKS

This paper presented a solution for homogeneous access on the Web to sensor generated data, for scientific research. By extending OGC service standards and implementing them as DCCs, our solution fosters interoperability in situations where geospatial sensor data need to be combined with other kinds of data sources, a common scenario in scientific research. Moreover, thanks to DCC construction principles, the solution supports posting of schematics (e.g., a workflow) to pre-process data before publication.

Ongoing work includes developing new kinds of DCC for implementing new mappings. We are also extending the DCC design and combination framework of (Santanchè et al., 2007) to support automated publication of DCCs as Web services.

ACKNOWLEDGEMENTS

This work is supported by FAPESP (grant 04/14052-3), CNPq, CAPES, and an HP Digital Publishing grant.

REFERENCES

Aloisio, G., Conte, D., Elefante, C., Marra, G. P., Mastrantonio, G., and Quarta, G. (2006). Globus Monitoring and Discovery Service and SensorML for Grid Sensor Networks. In *Proc. 15th IEEE WETICE'06*.

Chu, X., Kobialka, T., Durnota, B., and Buyya, R. (2006). Open Sensor Web Architecture: Core Services. In *4th Int. Conf. on Intelligent Sensing and Information Processing*, pages 98–103.

Gay, D., Levis, P., Behren, R., Welsh, M., Brewer, E., and Culler, D. (2003). The nesC language: A holistic approach to networked embedded systems. In *Proc. ACM Conf. on Programming Language Design and Implementation (PLDI'03)*.

Iamnitchi, Ripeanu, and Foster (2002). Locating Data in (Small-World?) Peer-to-Peer Scientific Collaborations. In *Proc. International Workshop on Peer-to-Peer Systems (IPTPS), LNCS*.

Levis, P., Madden, S., Polastre, J., Szewczyk, R., Whitehouse, K., Woo, A., Gay, D., Hill, J., Welsh, M., Brewer, E., and Culler, D. (2004). *Ambient Intelligence*, chapter TinyOS: An Operating System for Wireless Sensor Networks. Springer.

OGC (2007a). OpenGIS Reference Model (ORM). http://portal.opengeospatial.org/files/?artifact_id=3836 (as of Oct 2007).

OGC (2007b). OpenGIS Sensor Web Enablement Architecture. http:// www.opengeospatial.org/pt/14140 (as of Oct 2007).

Pastorello Jr, G. Z., Medeiros, C. B., and Santanchè, A. (2007). Providing Homogeneous Access for Sensor Data Management. Technical Report IC-07-012, Institute of Computing, UNICAMP.

Raskin, R. and Pan, M. (2003). Semantic Web for Earth and Environmental Terminology (SWEET). In *Workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data*.

Santanchè, A., Medeiros, C. B., and Pastorello Jr, G. Z. (2007). User-centered Multimedia Building Blocks. *Multimedia Systems J.*, 12(4):403–421.