An Event Driven Architecture for Decision Support

Thomas EVERDING and Theodor FOERSTER
Institute for Geoinformatics, University of Münster, Germany
{everdingt;theodor.foerster}@uni-muenster.de

Abstract. This paper presents an Event Driven Architecture for environmental monitoring and live decision support. Multiple OGC web services are integrated into this architecture, including the Web Processing Service and the Sensor Event Service. The system is demonstrated using a pilot from the EC funded GENESIS project. The architecture, the processing steps and the benefits of the system are described in detail.

Keywords. Event Driven Architecture, Spatial Data Infrastructure, OGC Web Services, Decision Support

1. Introduction

Environmental measurements are crucial for decision making and assessing health risks. The variety of measurements is accessed via Spatial Data Infrastructures (SDIs). However, the pure access to the information is not sufficient as the measurements have to be available in real-time. Especially phenomena with a strong effect on health and a fast spread have to be detected and processed as soon as possible.

This paper presents a near real-time system based on an Event Driven Architecture (EDA) and existing web services such as the OGC Web Processing Service (WPS) and OGC´s suite of sensor services. The EDA was developed as part of the FP-7 project GENESIS (GENeric European Sustainable Information Space for environment) [1] which builds collaborative information networks for environment management and health actors. It provides a Service Oriented Architecture (SOA) applying standards from various organizations like ISO, the OGC, the W3C and OASIS.

Section 2 describes basic concepts of OGC web service standards as applied in this paper. In Section 3 the use case applied to the presented system is described. The fourth section presents the system including its architecture and the utilized techniques and processes. Finally the benefits of the system are summarized and an outlook for future developments is given.

2. Related Work

This section provides an overview about the related work on EDAs and the sensor web as applied in this work.

For this paper we follow the event definition as developed in [2]: " *An event is anything that happens or is contemplated as happening at an instant or over an* *interval of time*". Such an event can represent the real world phenomenon or its digital representation. For this paper we understand an event as the latter. An Even Driven Architecture (EDA) is an architectural style in which most of the components execute their actions based on incoming events and communicate via events [3]. To extract specific information from events Event Stream Processing (ESP) is applied. Such processing is performed using so-called data views which provide access to a sub-set of the available events, e.g. all events received during the last hour or the newest 100 events. The processing rules are defined as event patterns [4, 5].

The Event Driven Architecture is based on standards for service interfaces and data encodings such as provided by the OGC for geographic applications [6]. The Web Processing Service (WPS) allows clients to perform configurable remote processes over the web [7]. Performing web-based processes on the web is regarded as the next step in SDIs and allows clients to build flexible processing chains [8, 9].

The Web Coverage Service (WCS) provides means to query coverage datasets over the web [10]. Coverages are for instance remote imagery or any n-dimensional raster dataset.

Besides the services for raster and processing, the Sensor Web Enablement (SWE) initiative at the OGC develops standards specific for sensors, sensor systems and sensor networks. In the recent years the SWE initiative released several service specifications as well as encodings for sensor metadata and for sensor measurements. Within the SWE initiative also services using a publish/subscribe communication were specified. These are namely the Sensor Alert Service (SAS) [11] and the Sensor Event Service (SES) [12]. The SAS allows clients to subscribe for sensor measurements using filter criteria like a bounding box or a threshold. The SES is a successor of the SAS, also based on the publish/subscribe messaging pattern. It makes use of existing standards such as the Web Services Notification (WS-N) suite from OASIS [13, 14, 15], Observations & Measurements [16] and the OGC Filter Encoding specification (FES) [17] for the definition of subscription filters. Additionally, it applies Event Stream Processing. The event patterns of the SES e.g. for Event Stream Processing are described through the Event Pattern Markup Language (EML) [18].

3. Use Case

The EDA presented in the following section is based on a use case of the GENESIS project about cyanobacterial bloom in the artificial Villerest reservoir. The reservoir is located in the north-west of Lyon, impounding the Loire River. In the summer months the reservoir is often affected by cyanobacterial blooming. During such a bloom phase the concentration of blue-green algae and natural toxins is largely increased to a degree that it even threatens humans. Consequently, drinking the reservoir's water as well as skin contact has to be avoided. The detection model for cyanobacterial bloom is based on in-situ measurements and remote imagery data [19, 20].

4. Event Driven Architecture

Based on the use case the presented EDA needs to perform processing of in-situ measurements (i.e. pH values) as well as of remote imagery. In the following the EDA with its components is described. In particular, the Enterprise Service Bus, the topic concept, the processing by the SES and the necessary extensions of the components are presented.

The components and data flows of the EDA are shown in Figure 1. The sensors provide pH value measurements. To encapsulate the sensors and to provide a sensor gateway these measurements are sent to a Sensor Alert Service (SAS). The SAS feeds them into the Enterprise Service Bus (ESB), a central communication middleware component, which serves as communication platform. All events (e.g. sensor measurements or other notifications) in the EDA are sent to the ESB and disseminated to the subscribers. The Sensor Event Service (SES), subscribed for pH value measurements, calculates the daily pH value variation based on the sensor measurements. In case of high variations, notifications are produced that trigger the Web Processing Service (WPS).

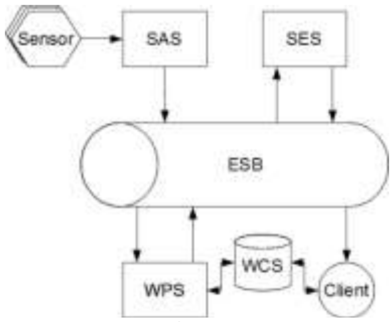

Figure 1. Overview of the components and data flows.

The WPS accesses remote imagery data for the area in which a high variation was detected and scans the image for high concentrations of Chlorophyll A. Details on algorithms for Chlorophyll detection can be found in [21]. The processing results are available through a Web Coverage Service (WCS) for querying. In addition, clients (e.g. a GIS or a web portal) are notified that new results are available. Based on the results, the user decides about restrictions in the affected water body.

4.1. Enterprise Service Bus and Topics

The Enterprise Service Bus handles all event communication. It provides a Web Services-Notification (WS-N) interface for notification as well as for subscriptions of specific notifications, which are grouped by topics. Topics allow users to subscribe for a type of notification without the need of knowing details like "measurements taken by sensor XY". They are abstract filter for types of notifications and are structured as a tree, where a topic may be a node containing further sub-topics or a leaf on which notifications are published. Clients can only subscribe for leaf topics.

Figure 2 shows the topics that were implemented for the use case. The SES for instance is subscribed for notifications on the 'PH' topic to receive every pH value measurement. Critical variations are published to the 'CriticalPHVariation' topic on which the WPS is subscribed. The distinction between 'Measurements' and 'DerivedInformation' allows separating the base data from information that is generated. This helps to identify the topics of interest, especially when taking further phenomena into account (e.g. oxygen saturation measurements and information derived thereof).

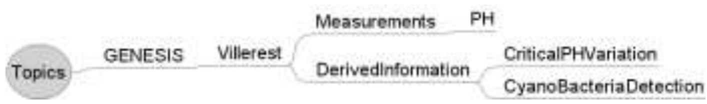

Figure 2. Overview of the topics.

4.2. Sensor Event Service Processing

The SES calculates the daily pH value variation and checks if it exceeds a critical threshold. According to [20], a critical pH value variation is one indicator for a cyanobacterial bloom. Table 1 gives the definition of the algorithm.

Table 1. Definition of the critical pH value variation detection algorithm [20].

Name	**Data Type**	**Unit**	**Value Range**	**Threshold for decision process**
pH	Float	pH Units	0 to 14	see below
pH variation	Float	pH Units / day	-14 to 14	abs(pH variation) >= 0.5
pH Status	Logical		Normal | Abnormal	if abs(pH variation) >= 0.5 then pH status = Abnormal, else pH Status = Normal

This algorithm is translated into Event Stream Processing (ESP) rules that can be executed on-the-fly in the SES (encoded as EML). For identifying cyanobacterial bloom five event patterns are created (see Figure 3).

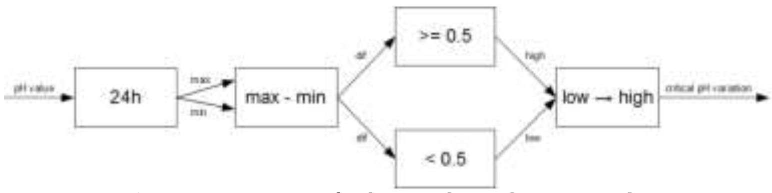

Figure 3. EML patterns for the critical pH value variation detection.

The pH values for a given location are passed onto the first pattern. It uses a data view to provide access to the measurements from the last 24 hours. From this view the maximum and the minimum values are selected and pushed to the next pattern. Here the difference is calculated which is the absolute value of the maximum pH value variation during the last 24 hours (see Table 1, thresholds). Next, this difference is forwarded to a pattern checking if the threshold of 0.5 pH units is exceeded.

In theory one could stop here and generate a notification to trigger the WPS process. But imagine a situation in which the pH value changed by more than 0.5 during the last 20 hours. In that case the three patterns described above generate an output for every new measurement received during the next four hours. This leads to needless triggering of the WPS as the critical situation happened earlier and was already recognized.

To avoid this behavior of the system, two more patterns are needed. A first one looks for pH value changes that are within an uncritical range (< 0.5). The results of the two threshold patterns are then given to the last pattern which checks if an uncritical variation (low) is followed by a critical one (high). If this criterion is matched, the threshold was exceeded for the first time, later critical changes are ignored. In order to generate an output again, the pH value variation has to drop to a normal state again.

4.3. Extensions of Components

Several adaptions to the components were required to execute the workflow described above. First, the Sensor Alert Service had to be connected to the Enterprise Service Bus. There were two options, either to change the SAS to provide WS-Notification compliant notifications or to extend the ESB to accept also notifications sent via XMPP, the messaging protocol used by the SAS. The second option was implemented as it allows an easier integration of external sensors provided via regular SAS instances. For this aim also Sensor Event Service had to be extended to be able to handle SAS notifications.

A larger modification was made at the Web Processing Service as it had to be connected via WS-Notification to the ESB. This connector is able to receive notifications from the SES, extract the location and further necessary information and build the request to execute the Chlorophyll A detection algorithm. Also the notifications after the finishing process are built by this connector and sent to the ESB.

5. Conclusion and Outlook

This article describes an Event Driven Architecture for environmental monitoring and health assessment. The presented architecture is based on standards for web services (e.g. WS-N, SES and WPS). The architecture has been applied to a use case based on the fresh water quality pilot C from the FP-7 project GENESIS.

The presented architecture shows several benefits of an Event Driven Architecture compared to a classic Service Oriented Architecture. At first, the system works on-the-fly, which means that processes are started as soon as possible, namely as the input data (e.g. in case of the SES) or the trigger (e.g. in case of the WPS) is available. This way, the decision maker gets notified when a new decision is needed instead of guessing when new information for the decision has to be requested. Furthermore, there is no need for frequently requesting this inputs which may lead to unnecessary requests (if the frequency is too high) or to missed information (if the frequency is too low). This makes the system very flexible in terms of the application area and deployment context.

This architecture is also very flexible. Due to the loose coupling of the components, data sources can be exchanged easily. New sensors can be added or other remote imagery stores can be used during run time. Also service instances can be added or exchanged without stopping the system. This allows to migrate services to more powerful machines, to add redundant backup machines to avoid down times or to add new services to extend the workflow, e.g. by decision support services that automatically provide recommendations on necessary actions based on the WPS processing results. Furthermore, the processing rules and algorithms of the WPS and SES can be changed during run time by altering the subscribe (SES) or execute (WPS) requests. This might for instance be necessary due to a change of the remote imagery source. From all this also follows that the system needs to be set up and initialized only once. From that point on it will automatically execute the whole workflow.

Using an Enterprise Service Bus as the communication infrastructure also provides a single point to access data. This might be the final WPS processing results but also the intermediate results of the SES or the SAS notifications.

To make use of these benefits it is envisioned to integrate event based communication mechanisms into existing Spatial Data Infrastructures (SDIs) rather than to replace the current request-response based technologies. This was also used in the EDA presented in this paper: the remote imagery was not sent via the ESB but stored in a Web Coverage Service; only notifications referring to new processing results were distributed this way. This approach is also followed by the newly formed Pub/Sub Standards Working Group (SWG) of the OGC. It develops a standard for publish/subscribe usable by existing OGC web services.

Acknowledgement

This work was supported by the European Commission through the GENESIS project, an Integrated Project, contract number 223996. Special thanks go to the project partners that helped to implement the presented EDA.

References

[1] GENESIS Consortium, *Project overview*, 2010, last accessed on 28.01.2011. [Online]. Available: http://www.genesis-fp7.eu/index.php/project-overview

[2] T. Everding and J. Echterhoff, *OGC OWS-6 SWE Event Architecture Engineering Report*, Open Geospatial Consortium, Tech. Rep. OGC document number 09-032, 2009.

[3] D. Luckham and R. Schulte, *Event processing glossary - version 1.1*, Event Processing Technical Society, Tech. Rep., 2008.

[4] D. Luckham, *The Power of Events*. Addison-Wesley, 2002.

[5] O. Etzion and P. Niblett, *Event Processing in Action*. Manning Publications Co., 2010.

[6] Open Geospatial Consortium, *About OGC*, 2010, last accessed on 28.01.2011. [Online]. Available: http://www.opengeospatial.org/ogc

[7] P. Schut, *OpenGIS Web Processing Service*, OGC document number 05-007r7, Open Geospatial Consortium Std., 2007.

[8] C. Kiehle, K. Greve, and C. Heier, *Standardized geoprocessing - taking spatial data infrastructures one step further*, in 9th AGILE International Conference on Geographic Information Science, 2006, pp. 273–282.

[9] B. Schaeffer and T. Foerster, *A client for distributed geo-processing and workflow design*, Journal for Location Based Services, vol. 2(3), pp. 194–210, 2008.

[10] A. Whiteside and J. D. Evans, *Web Coverage Service (WCS) Implementation Standard*, OGC document number 07-067r5, Open Geospatial Consortium Std., 2008.

[11] I. Simonis and J. Echterhoff, *OGC Sensor Alert Service Implementation Specification*, OGC document number 06-028r3, Open Geospatial Consortium, Tech. Rep., 2007.

[12] J. Echterhoff and T. Everding, *OpenGIS Sensor Event Service Interface Specification (proposed)*, Open Geospatial Consortium, Tech. Rep. OGC document number 08-133, 2008.

[13] S. Graham, D. Hull, and B. Murray, *Web Services Base Notification 1.3*, OASIS Std., 2006.

[14] D. Chapell and L. Liu, *Web Services Brokered Notification 1.3*, OASIS Std., 2006.

[15] W. Vamberge, S. Graham, and P. Niblett, *Web Services Topics 1.3*, OASIS Std., 2006.

[16] S. Cox, *Observations and Measurements - Part 1 - Observation schema*, OGC document number 07-022r1, Open Geospatial Consortium Std., 2007.

[17] P. A. Vretanos, *OpenGIS Filter Encoding Implementation Specification Version 1.1.0*, OGC document number 04-095, Open Geospatial Consortium Std., 2005.

[18] T. Everding and J. Echterhoff, *Event Pattern Markup Language (EML)*, Open Geospatial Consortium, Tech. Rep. OGC document number 08-132, 2008.

[19] M. Gerbaux, *Fresh water quality state of the art report*, GENESIS, Tech. Rep. 3.0, 2010.

[20] ——, *Fresh water quality thematic pilot specification document*, GENESIS, Tech. Rep. 1.0, 2009.

[21] M. M. Islam and C. W. Tat, *A comparison of empirical algorithms for chlorophyll concentration in singapore regional waters*, in Proceedings of the 22nd Asian Conference on Remote Sensing, Singapore, November 2001.