Applying Complex Event Processing and Extending Sensor Web Enablement to a Health Care Sensor Network Architecture

Gavin E. Churcher and Jeff Foley

BT Research Adastral Park,
Martlesham Heath, Ipswich IP5 3RE
{gavin.churcher,jeff.foley}@bt.com

Abstract. The limited reuse of middleware components for wireless sensor networking projects has driven interest in emerging standards from the Sensor Web Enablement Working Group which offers methods to virtualize sensor data into a common, self-describing format, using access mechanisms based on HTTP. Using these standards, applications are able to discover and access different sensor offerings, automatically understand the data format used and even specify conditions in the sensor data. This paper examines how an existing sensor network platform in the health care domain can make use of these standards and examines the possibility of extending the Sensor Alert Service with a richer set of functions. Concepts taken from Complex Event Processing engines are explored in the context of this particular health care platform, where it is shown that there are clear advantages to extending the standard.

Keywords: Complex Event Processing, Health Care, Sensor Web Enablement.

1 Introduction

The Sensor Network Group in BT Research has had a long interest in wireless sensor networking based projects and has participated in a number of collaborations covering a wide range of domains from health care and assisted living through to environmental monitoring, and traffic management. Typically, these projects have required bespoke solutions where middleware reuse has been minimal. Our interest in emerging standards to address this issue has led us to investigate the use of Sensor Web Enablement (SWE) from the Open Geospatial Consortium (OGC) [1] as a possible approach for the virtualization of sensors. This would form part of our strategy for building a more generic sensor network architecture with components that could be reused in a diverse range of sensor network projects. We feel that the use of standardized middleware will help drive the acceptance of sensor network solutions as we move away from costly, bespoke solutions; a problem particularly relevant to our approach given the broad range of application areas.

We have investigated the SWE standards for the Sensor Observation Service (SOS) [2] and Sensor Alert Service (SAS) and how they may be applied to one of our existing sensor network projects, SAPHE (Smart and Aware Pervasive Healthcare Environment) [3]. SAPHE is one of a number of growing industrial and academic

S. Hailes, S. Sicari, and G. Roussos (Eds.): S-Cube 2009, LNICST 24, pp. 1–10, 2009.

wireless sensor network projects in the health care domain. The most notable being Intel IrisNet [4], Hitachi Collectlo [5], and A Remote Health Care System Based on Wireless Sensor Networks [6].

This paper reviews the previously published findings where we applied SWE to SAPHE [7] and highlights the possibilities of extending the SAS service with more advanced filtering mechanisms, such as those found in Complex Event Processing (CEP). An example CEP engine, Esper [9] was investigated with the view that the application of CEP to the SAS service may result in a number of benefits to sensor network architectures and to SAPHE in particular. CEP is particularly relevant to SAS as it offers the ability to aggregate and correlate large volumes of events through the real-time processing of continuous queries. It is possible to apply pattern matching to asynchronous events through the use of logical and temporal event correlation, and defined 'window' views of the event streams. Current standards for defining which events are relevant to an application using SAS are severely limited to a simple definition of a property of a single event. It is not possible to correlate multiple events, a capability that has the potential to reduce bandwidth and processing overheads of edge network applications.

2 Sensor Web Enablement

The concept of Sensor Webs was coined in the 1990's: millions of connected on-line sensors monitoring the physical world, the sensor capabilities being described using metadata so they can be published and understood by anyone with web access and appropriate authentication. This model is similar in concept to the World Wide Web where a standard web browser can access this vast information space due to the adoption of key standards such as HTTP, HTML and XML. In 2001, a data modeling language for sensors, SensorML [10], was introduced into the OGC which led to the SWE working group. The group was tasked to produce a framework of open standards for Web connected sensors and all types of sensor systems. The SWE standards draw from a number of existing OGC standards from SensorML to Observations & Measurements [11] and propose a number of services that use the HTTP protocol for access. The services are able to self-describe the data they represent and the access mechanisms they provide.

Version 1.0 of the Sensor Observation Service (SOS) was published as an OGC OpenGIS Implementation Standard in October 2007 [2]. Of particular interest to SAPHE are the proposed standards for SOS and SAS. Together they provide the ability to store data that can be accessed and queried by an external application and to detect simple conditions that can then generate an alert, published to the external network and various application subscribers.

The role of the SOS is to translate incoming sensor data into a data model that represents the sensor and the data as a series of observations of a particular feature of interest. This translated representation is then archived until an external client makes a request for data, providing a number of parameters that act as filters. The parameters for the SOS include the ability to specify a time constraint, for example, between two time periods, or after a time point, and the identification of the particular sensor cluster. In the case of multiple sets of data from a sensor cluster, the observed property (phenomenon) can be specified along with the features of interest. Once the request for data has been made, the application client receives a set of Observations and Measurements which encapsulate the sensor data in the data model. Sensor data can also be streamed into an SAS where a similar transformation occurs. An external application is able to specify conditions for data as it arrives at the SAS. When the data meets those conditions an event is sent to the application using the publisher/subscriber methodology. The conditions that can be applied in the current proposal for the SAS are limited to specifying whether a sensor data value is less than (or equal to), greater than (or equal to), equal to, not equal to a value, or between two values. For example, an application could specify that an alert should be sent when a PIR sensor detects movement, or when a temperature sensor reports a value over 40ºC which could be critical for certain medicines.

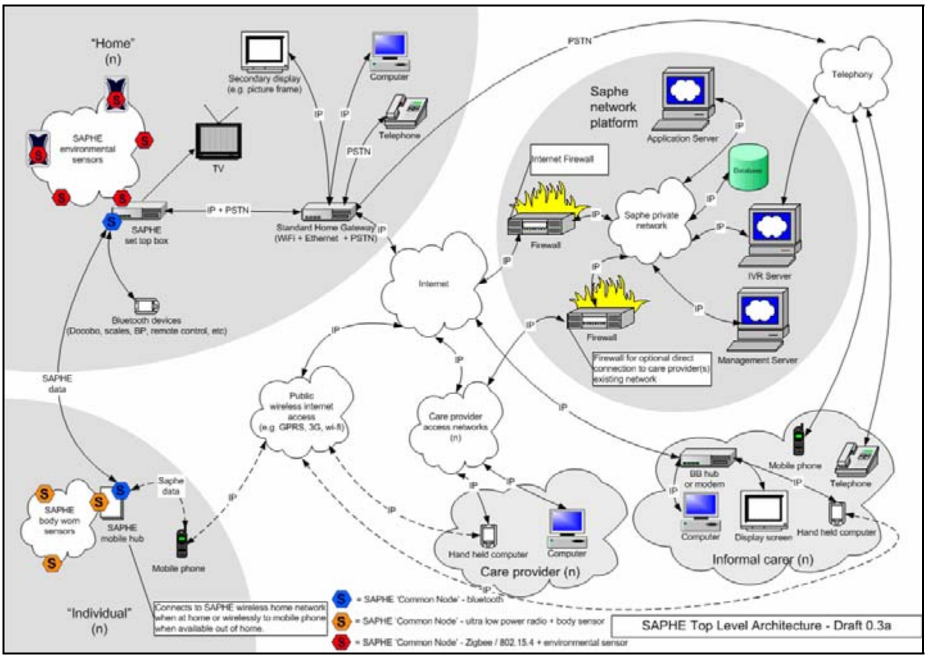

Fig. 1. SAPHE high level architecture

3 Key Challenges in SAPHE

SAPHE is a collaborative research project co-funded by the UK's Technology Strategy Board, involving Imperial College, BT, Philips UK, University of Dundee, Cardionetics and Docobo. It aims to develop a holistic monitoring solution to support the care and self-care of people with long-term health conditions with the placement of a number of sensors around the home and on the patient that monitor both the immediate environment in the home along with physiological traits. SAPHE targets patients who typically receive a specialist service provided by community matrons and multidisciplinary teams. The desire is to support these users, providing a new tool for professional care, encouraging greater patient self-care and to monitor the patient in order to detect early indications that a patient's wellbeing is changing and preventative care is required. Independent monitoring of the patient and their environment can lead to early detection of worsening conditions that may either not be reported by the patient nor detected by the health care professional and help prevent escalation of a patient's conditions and their ability for self-care. For example, changes in sleeping patterns, mobility around and outside of the home, and eating habits can be early indicators of a worsening of a patient's condition.

The present SAPHE system architecture is shown in Fig. 1. Within the home environment there are a number of sensors that use either ZigBee or Bluetooth to communicate to the SAPHE set-top box which has the task of managing the communication from the sensors and reformatting the sensor data into a common format based on BinX [12]. The canonized data is then sent securely to the SAPHE network platform via the Internet, using the BT Home Hub as a gateway. Within this network the data is analyzed for significant events and other factors that can lead to an assessment of the patient's wellbeing. This information is then sent on to health care professionals. For example, patient data is visualized in real-time on the secure SAPHE health care monitoring portal as a series of histograms or line graphs using Dundas Chart, which the authors helped to develop and is shown in Fig. 2.

The patient also wears a number of sensors in the form of a single device worn on the ear. This device reports back blood oxygen levels, heart rate and an activity index derived from a 3D accelerometer when within range of the set-top box within the home. When outside the home environment, the patient wears a mobile device that stores the body-worn sensor data until it is in range of the set-top box at which point it uploads all cached data. The mobile device is also able to communicate directly with the SAPHE network platform in the case of an emergency via a Bluetooth connection to the patient's mobile phone and GPRS connection.

The SAPHE environmental and physiological sensors and the observations they report are listed in Table 1, below.

Table 1. SAPHE Sensors

Sensor	Reports	Comms	Message Frequency
PIR room sensors	Detection of movement within range of sensor	ZigBee	Up to 10,000 per 24h period
Entry/exit door sensors	door open/close event	ZigBee	Varies according to patient
Fridge door sensor	door open/close event	ZigBee	Varies according to patient
Activity sensor	activity index, SpO2 and bpm	Proprietary low-power radio	Aggregate 1 per second
Weighing scales	weight when used	Bluetooth	2 per day
Blood pressure monitor	blood pressure when used	Bluetooth	2 per day
Bed sensor	number of bed exits, first time in bed, last time out of bed	Bluetooth	Every 30 seconds

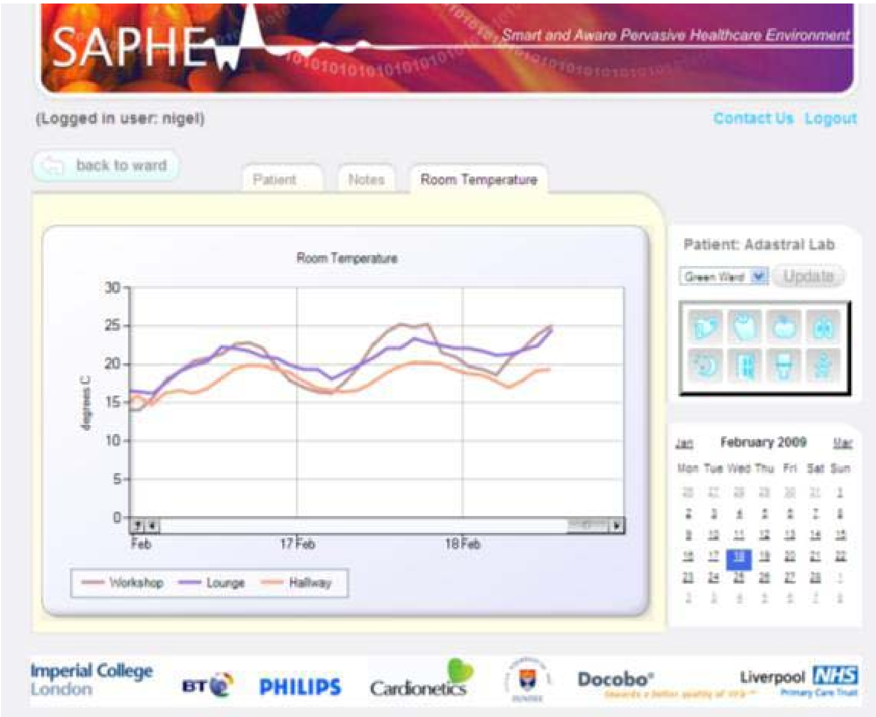

Fig. 2. SAPHE monitoring portal showing room temperature sensor data

4 Lessons Learned

In the current proposed SAPHE architecture, all sensor data from each patient and home environment is sent to the external, back-end servers in the SAPHE platform network which archive and check for patterns and trends in the data that are indicators of deteriorating health in the patient. The frequency of sensor communication and the overhead of BinX sent externally would indicate large volumes of bandwidth usage growing as the patient user base expands.

Creating SWE services offers a number of generic advantages and some specific to this type of application where local processing could prove advantageous. SWE offers a standardized protocol for discovering and accessing sensor data which enables data to be reused in potentially new and novel ways. An application could simply repurpose sensor data for another domain, or fuse together data from several services to provide radically different applications. Standardizing on the access mechanism and the data model for the sensor data conveys advantages to the application developers as there are a growing number of 3rd party tools that facilitate access, analysis and visualization of data, reducing the time to develop new applications and facilitating innovation [13].

SWE services can exist anywhere in the architecture between the sensors and the applications that utilize their data. Specific to SAPHE and similar sensor network architectures, placing the SWE services at the local level could reduce real-time bandwidth requirements. An SOS archives sensor data in a common data format allowing applications to query and retrieve data as appropriate. An SAS would be able to offer basic analysis of sensor data, publishing an alert or data fragment to subscribing applications. An application could then use the SOS to access relevant data when appropriate rather than receiving data in real-time for analysis.

The architecture and protocols were already established for the SAPHE system, making use of proprietary data formats and protocols. The SWE services were created in parallel to the existing framework, an approach not untypical for sensor network platforms where such services can readily be developed as an adjunct to an existing platform; in effect, applications can be retro-fitted to provide SWE services. Our previous research [7] shows how an SOS service could be retro-fitted to the existing framework. The exercise was a valuable one and clearly showed that there was quite a high overhead in creating a standards-based service. The cost in doing so would hopefully, with time, be mitigated through the reuse of that service by new applications, although it remains to be seen whether the cost would be simply too high for closed solutions.

The process of creating an SOS service began with the definition of the data models that represented the sensor data in terms of 'observations' of 'features of interest' that were presented in the form of 'offerings'. There were two initial types of offering: sensor data from the ambient sensors around the house, and body-worn physiological sensor data, neatly mapping on to two features of interest being the house and the patient. Sensors and sensor clusters are known as 'procedures' under SWE terminology. A simple example of how the data model would look for a set of single sensor observations follows:

Table 2. SAPHE Data Model

Offering:	Physiological observations
Feature of Interest:	Patient
Phenomenon:	Heart rate
Procedure:	Cardionetics ECG Sensor
Unit:	Beats per minute

The process of creating an SAS overlaps with that of creating the SOS since the definition of the data models and the transformation from raw sensor data to these models is the same. The SWE standards do not stipulate how the services should be implemented and so an SOS and SAS can exist on the same extended platform, receiving the same sensor data and even sharing the data transformation overhead. The SAS provides a service where an external application can specify the sensor data conditions which would lead to an alert being generated and published to that subscribing application. The contents of that alert could be simple message or actual sensor data. In this discussion we consider the arrival of sensor data at a service to be an event.

The role of an SAS as defined in the current proposed specification would be of limited benefit to the SAPHE platform because of the narrow range of conditions that can be tested for in the filter schema. The SAPHE system relies on the detection of patterns in the sensor data often over differing periods of time. One main limitation of the current filter specification is the inability to aggregate sensor readings over time. The data collected from both body-worn and ambient house sensors provide a rich picture of a patient's activity and wellbeing. The notion of wellbeing depends on the context of the individual and their own patterns of behaviour. Detecting deviation from the norm in certain behaviours, for example increased activity in the night indicative of a deteriorating sleep pattern or a drop in consumption of food and water, requires specialist applications that perform statistical analysis on large volumes of data.

An SAS functioning at the local level could still be of benefit to SAPHE if it could be extended to handle more sophisticated conditions on the sensor data. The ability to examine sensor data for patterns of behaviour and create appropriate abstract events that can be published to applications could lead to the advantages of SWE being realized in SAPHE and reduce the bandwidth needed for real-time communication of sensor data to applications.

Complex Event Processing (CEP) is an event processing concept that takes asynchronous, real-time, high-volume data event streams and provides a mechanism for application developers to specify correlations, aggregations and other forms of event pattern matching. The approach taken by CEP turns the traditional, database-led approach of application development upside-down. Rather than an application repeatedly compiling a query, submitting it to a database and waiting for a result, applications using CEP submit a query once. This is compiled by the engine and as data events arrive they are passed through this query. When conditions are met, the resulting data is published to the subscribing application. CEP provides a publish/subscribe view of event streams that supports complex analysis of the data stream and negates the need for an application to repeatedly poll a database.

Typically an application registers one or more queries that are similar in style to SQL but have been extended to support the correlation and pattern matching of asynchronous events. Pattern matching for instance, supports the occurrence of sequences of events meeting certain criteria, and even detect the non-occurrence of events. The versatility of CEP to specify correlations and analysis of data streams, makes it potentially a very useful component to use in the analysis of sensor data and applicable to a wide range of wireless sensor networking applications.

From the small number of CEP engines available we chose Esper [9], a Java and .NET based framework because of its extensive documentation, online community support and open source licensing. Esper supports many of the critical functions needed by CEP applications that require low-latency analysis of real-time data. Esper supports the following key methods of analysis in CEP:

- windows on events: sliding windows (time, length, sorted, time-ordered); tumbling windows (time, length, multi-policy, first-event)
- grouping, aggregation, sorting, filtering and merging of event streams
- output rate limiting and stabilizing
- access to a wide range of data formats using a standardized interface language
- logical and temporal event correlation

One of the basic, yet powerful ways of using a CEP engine is to define a pattern. These examples are taken from [9] and use the EPL language to define rules. Programmatic handlers can detect when a pattern has a match and report back to the CEP container/application. The following is an example of a time-based pattern where after event ‘A’ arrives it will wait 10 seconds before reporting:

```
A -> timer:interval(10 seconds)
```

More sophisticated patterns using sequences and time windows can be easily expressed, for example the following detects event ‘A’ followed by event ‘B’. Once ‘B’ is found then reset the pattern:

```
every ( A -> B )
```

Patterns can be combined with SQL-style SELECT statements to create increasingly sophisticated rules, for example the following taken from [9] will look for the occurrence of three temperature sensor events that report a temperature of more than 50 degrees within 90 seconds of the first event, with no events reporting a reading below that threshold. This pattern is inserted into another internal stream upon which other rules can be based. Chaining of rules can lead to sophisticated pattern matching.

```
insert into TemperatureWarning
select * from pattern
[every sample=Sample(temp > 50) ->
((Sample(sensor=sample.sensor, temp > 50) and not Sam-
ple(sensor=sample.sensor, temp <= 50))
->
(Sample(sensor=sample.sensor, temp > 50) and not Sam-
ple(sensor=sample.sensor, temp <= 50))
) where timer:within(90 seconds))]
```

There are a number of design patterns for applications that analyze asynchronous, real-time, high-volume event streams.

Within SAPHE there is the need for applications to abstract away from the raw sensor data and look for patterns which could indicate certain events have occurred. These events could then form the basis for further statistical analysis, contributing to a broad picture of a patient’s wellbeing and the detection of early symptoms of a deteriorating situation. One factor of a patient’s wellbeing relates to how sociable the patient is, for example, how often they leave their house, or whether they have visitors on a regular basis. A good example of this is automatically detecting when there is more than one person in the patient’s house which can then form the basis for more sophisticated analysis. The PIR sensors in each room send data whenever movement is detected, potentially up to 10,000 events per 24 hour period per sensor. Providing the logic to look for meaningful events such as multiple occupancy or an empty house from this at a local level would negate the need to transmit the raw sensor data to the back-end applications. The logic to detect multiple occupancy could be represented as follows:

if a PIR sensor (PIR_A) reports movement and a different PIR sensor (PIR_B) reports movement within 5 seconds of the first, then report multiple occupancy

This simple example can be extended much further where sensors in non-adjacent rooms are triggered within a specified time-frame, or combined with other sensors such as the bed activity sensor and the front-door. The ability to specify a window of time from which to look for patterns in the data is essential to detect these higher-level, application specific events. CEP is adeptly suited to detect these types of events from the real-time data. CEP engines such as Esper also provide statistical analysis of patterns of events.

An SAS based on the current proposed standard could readily be implemented using a CEP engine, however it would be unable to take advantage of the level of sophistication possible in CEP, particularly the ability to analyze across a number of sensor data readings. The SAPHE platform needs to perform high-level correlations and analysis of sensor data in order to calculate critical factors for a patient. Some of this correlation and analysis could be performed 'in-network' using CEP as opposed to at the 'edge'. Exposing the rich functionality of a CEP engine through an SAS would convey the advantages of both – expressiveness in pattern matching alongside access to data and its derivatives through a standard protocol. The ability to process this data in-network also has a tangible benefit to the network bandwidth and processing overhead of the edge SAPHE applications. As the application domain scales up the number of sensors and patients, placing processing close to the data source will lead to lower overheads and a more rapid response from a system that is critical to the welfare of its patients.

5 Conclusions and Future Work

This paper has reviewed how Sensor Web Enablement services can be retro-fitted to an existing sensor network platform and has highlighted what the potential benefits are in doing so. SWE enables sensor data to be virtualized, providing a common, self-describing data format and access protocol. The number of domain-agnostic toolkits becoming available indicates that the rather large overhead in creating new applications based on accessing these services can be mitigated by the re-use of data, the use of third-party analysis engines and the reductions in bandwidth and processing overhead to edge applications. The ability to access a diverse range of real-time data has the potential to lead to exciting and radically different applications including health care.

Considering the range and growing number of sensors monitoring each patient and his or her environment, there is a recognized need to optimize the processing of sensor data in order to make informed inferences on the well-being of each patient. Support for data fusion using components from the SWE framework (e.g. SAS) extended through concepts such as CEP may prove to be a valid approach to meeting this growing volume and complexity of data whilst providing a standard method for accessing this data.

Technologies such as Complex Event Processing are designed to process high-volumes of sensor data with minimal latency. They provide a potential solution to the growing world of sensor data that is becoming available. Our experiences with Esper highlights that CEP is ideal for this critical and dynamic environment in contrast to a traditional database approach, where real-time processing of large volumes of data is critical. With respect to the SAPHE project, we have shown that in this and previous research, it is possible to retrofit existing wireless sensor network projects with SWE services.

Recent events have seen the publication of two OGC discussion papers proposing the adoption of Event Pattern Markup Language (EML) [14] and OpenGIS Sensor Event Service Interface Specification [15] for SWE services and in particular SAS. These approaches continue the discussion on the need for a more flexible and extensible method of defining which events and sequences of events are of interest to edge applications. The exercise of applying SWE to SAPHE has added to that discussion and the potential benefits of using a CEP-style aggregation/correlation engine made clear.

Acknowledgements

This research was supported by British Telecommunications Plc., University College London and the EPSRC. Our thanks to J. Echterhoff (iGSI) for SAS developments, T. Mizutani (BT) for SAPHE sensor capabilities, and Dr. Yang (UCL) for suggested revisions.

References

1. Botts, M., Percivall, G., Reed, C., Davidson, J.: OGC Sensor Web Enablement: Overview and High Level Architecture. OGC Inc. 06-050r2 (2006)
2. Na, A., Priest, M.: Sensor Observation Service. OGC Inc. 06-009r6 (2007)
3. Barnes, N., Mizutani, T., et al.: SAPHE Architecture Overview (2008), `http://ubimon.doc.ic.ac.uk/saphe/m338.html`
4. Gibbons, P.B., Carp, B., Ke, Y., Nath, S., Seshan, S.: IrisNet: An Architecture for a Worldwide Sensor Web. In: Pervasive Computing. IEEE, Los Alamitos (2003)
5. Ando, N.: Sensor Information Web Service for Healthcare Management at Home Powered by Collectlo. Hitachi (2008)
6. Zhang, P., Chen, M.: A Remote Health Care System Based on Wireless Sensor Networks. IEEE Xplore (2008)
7. Churcher, G., Foley, J., Bilchev, G., et al.: Experiences Applying Sensor Web Enablement to a Practical Telecare Application. In: ISWPC, Greece (2008)
8. Foley, J., Churcher, G.: Recent Developments in the Design of Sensor Network Architectures. In: 2nd European Conference on Smart Sensing and Context, England (2007)
9. EsperTech: Esper Reference Documentation, Version 2.2.0, `http://esper.codehaus.org/`
10. Botts, M.: Sensor Model Language for In-situ and Remote Sensors. OGC Inc. 02-026r4 (2002)
11. Cox, S.: Observations and Measurements. OGC Inc. 05-087r4 (2006)
12. Binary XML Description Language, `http://www.edikt.org/binx`
13. 52North OX-Framework, `http://52north.org/`
14. Everding, T., Echterhoff, J.: Event Pattern Markup Language 08-132 (2008)
15. Echterhoff, J., Everding, T.: OpenGIS Sensor Event Service Interface Specification 08-133 (2008)