SEMANTIC SENSOR NETWORKS

Proceedings of the 4th International Workshop on Semantic Sensor Networks 2011 (SSN11)

Bonn, Germany, 23 October 2011

A workshop of the 10th International Semantic Web Conference (ISWC 2011)

Kerry Taylor, Arun Ayyagari and David De Roure, Editors

Sponsored by

EU Project Spitfire, contract no. 2588 http://spitfire-project.eu/

CSIRO Australia http://www.csiro.au/

Preprint. To be published in CEUR-WS proceedings http://CEUR-WS.org

Program Committee

Arun Ayyagari	The Boeing Company
Franz Baader	TU Dresden
Luis Bermudez	Open Geospatial Consortium
Boyan Brodaric	Geological Survey of Canada
Mark Cameron	CSIRO ICT Centre
Michael Compton	CSIRO ICT Centre
Oscar Corcho	Universidad Politécnica de Madrid
David De Roure	Oxford e-Research Centre, University of Oxford
Ralf Denzer	Saarland State University for Applied Sciences (HTW)
Peter Edwards	University of Aberdeen
Alasdair Gray	University of Manchester
Manfred Hauswirth	Digital Enterprise Research Institute (DERI), Galway
Cory Henson	Kno.e.sis Center, Wright State University
Krzysztof Janowicz	GeoVISTA Center, Department of Geography; Pennsylvania State University, USA
Laurent Lefort	CSIRO ICT Centre
Yong Liu	NCSA
Kirk Martinez	University of Southampton
Thomas Meyer	UKZN/CSIR Meraka Centre for Artificial Intelligence Research
Andriy Nikolov	Knowledge Media Institute, The Open University
Kevin Page	Oxford e-Research Centre, University of Oxford
Josiane Parreira	Digital Enterprise Research Institute (DERI), National University of Ireland, Galway
Sascha Schlobinski	cismet GmbH
Amit Sheth	Kno.e.sis Center, Wright State University
Ingo Simonis	International Geospatial Services Institute
Kerry Taylor	CSIRO ICT Centre

Table of Contents

Automated context learning in ubiquitous computing environments

Fano Ramparany[1], Yazid Benazzouz[1], Jérémie Gadeyne[1], and Philippe Beaune[2]

[1] Orange Labs
Meylan, France
fano.ramparany@orange-ftgroup.com
[2] Ecole Nationale Supérieure des Mines de St-Etienne
St-Etienne, France
philippe.beaune@emse.fr

Abstract. Context awareness enables services and applications to adapt their behaviour to the current situation for the benefit of their users. It is considered as a key technology within the IT industry, for its potential to provide a significant competitive advantage to services providers and to give subtantial differentiation among existing services.
Automated learning of contexts will improve the efficiency of Context Aware Services (CAS) development. In this paper we present a system which supports storing, analyzing and exploiting an history of sensors and equipments data collected over time, using data mining techniques and tools. This approach allows us to identify parameters (context dimensions), that are relevant to adapt a service, to identify contexts that needs to be distinguished, and finally to identify adaptation models for CAS such as the one which would automatically switch off/on of lights when needed.
In this paper, we introduce our approach and describe the architecture of our system which implements this approach. We then presents the results obtained when applied on a simple but realistic scenario of a person moving around in her flat. For instance the corresponding dataset has been produced by devices such as white goods equipment, lights and mobile terminal based sensors which we can retrieve the location, position and posture of its owner from.
The method is able to detect recurring patterns. For instance, all patterns found were relevant for automating the control (switching on/off) of the light in the room the person is located. We discuss further these results, position our work with respect to work done elsewhere and conclude with some perspectives.

1 Introduction

Context awareness is considered as a key technology within the IT industry, for its potential to provide a significant competitive advantage to services providers and to give subtantial differentiation among existing services. According to a Gartner Inc. report [1], "Context-aware computing today stands where search engines and the web did in 1990".

In parallel to this, the interest of the scientific community in the context aware computing domain has gained a lot of momentum, due to the fact that with the advent of the

Internet of Thing (IoT) era, terabytes of data are bound to be produced daily by sensors and equipments.

Such data, when correctly interpreted can enrich the description of the context, which in turn makes it possible for services and applications to get context-aware, and finally to improve their efficiency in terms of personalization, and simplicity of use.

However, identifying and describing/defining relevant contexts is cumbersome. One reason is that it is generally the case that multiple contexts have to be identified and distinguished. Another is that contexts span over multiple domains such as the “user context”, the “system context” or the “environmental context”, to mention only a few.

Thus, the automated learning of contexts is a way to improve the efficiency of Context Aware Services (CAS) development.

Our approach consists of storing, analyzing and exploiting an history of sensors and equipments data collected over time. In a previous work we have used a semantic modeling language for describing context information [2] and have proved that semantic modeling makes it possible to describe heterogeneous information in a single framework. More generally, interoperability among sensors, sensors networks, and sensor based applications has been promoted by initiatives such as the Semantic Sensor Network incubation group (SSN) [3]. In the work reported here, weve sticked to that semantic modeling policy. As explained throughout this paper, this will allow us to:

- Identify parameters (context dimensions), that are relevant to adapt a service, such as the control of lights or white goods equipment. For example, the user activity is such a parameter and the next item gives an example on how this parameter is used to define contexts.
- Identify contexts that needs to be distinguished. For example, if I need more light when I read than when I watch the television, the context “I am reading” should definetely be distinguished from the context “I am watching the television”. Both contexts refer to my activity and going back to the previous item, the activity should be identified as a parameter that is relevant to our concern.
- Identify adaptation models for CAS such as the one which would automatically switching off/on of lights when needed

In the next section we introduce a simple scenario, which will illustrate a standard use case that our system supports. The details of the scenario will be used throughout the paper to provide concrete examples of the concepts involved in our approach. We then present our approach and describe the architecture of our system which implements it. The system has then been assessed on several datasets. We present the results obtained when applied on the illustrative scenario dataset. Finally, we discuss these results and position our work with respect to work done elsewhere and conclude with some perspectives.

2 Jane ordinary day life Scenario

The scenario takes place in a simple flat and stages Jane, a 80-year-old lady who spends the first two hours of the day moving back and forth between her bedroom and her kitchen. The map of the flat is depicted in figure 5-(a). More precisely, at the beginning of the scenario, Jane is sleeping in her bed, then she wakes up, goes to the kitchen, eventually she uses her oven to bake or reheat some food, eats it and then returns to her bedroom to take a short nap. Then she walks back to the kitchen to drink a glass of water and returns again in her bed to resume her short rest.

The flat is equiped with a sensor which keeps track of the status of the oven, i.e. if the oven is on or off, and with lights which emit signals whenever they are turned on and turned off. These devices and sensors are also pictured in in figure 5-(a). Jane keeps her mobile phone with her. The mobile phone embeds a software which is able to detect Jane's location, i.e. whether she is in her bedroom or in her kitchen. It also embeds a software which is able to detect Jane's posture, i.e. whether she is lying, standing, seating or walking.

Now by observing Jane's behaviour over a long period of time, say over a week, a human would probably notice that most of the time, if not everytime, when Jane wakes up and gets out of her bed she switches the light on, and that most of the time when Jane leaves her bedroom she switches the light off. Our claim is that we could achieve a similar analysis by applying data mining techniques on a corpus of sensors data, correlated with Jane behaviour, and collected over the same period of time.

Actually, we believe that modeling the sensors data using an appropriate representation language, storing them over time in a database and analyzing the content of this database using datamining techniques, will make it possible to discover contexts which might be relevant for adapting services in such a way that they would be personalized to Jane.

We elaborate this and introduce our approach in the following section.

3 Approach and architecture

The notion of Context is itself contextual as each application, each user, each activity has its own definition of context. For this reason there's no point considering a monolithic or centralized context management system. This lead us to opt for a context management infrastructure that each party could use to setup and manage its own context, rather than for a central context management system, which implicitely would mean that some universal contexts exists that would suit to all parties.

Moreover, the architecture as well as the information model should be flexible. More precisely, the modeling language should be able to cope with the heterogeneity of data sources as well as with the variety of nature of data produced by these data sources. For all these reasons we have based our approach on the Amigo Context Management Service (CMS)[4]. We recall here the main concepts of this framework. For more details the reader could refer to [4].

Each sensor or data source is encapsulated within a software component that we call a context source (CS). An example of this is depicted in the figure 1 where a mobile phone using Wifi based location feeds a software component called "location CS".

The connection between real sensors and its CS component is dependent on the sensor connectivity. In principle, all options can be supported, among which, the most popular ones are the serial line, PLC, Zigbee, ethernet, bluetooth connectivities. The

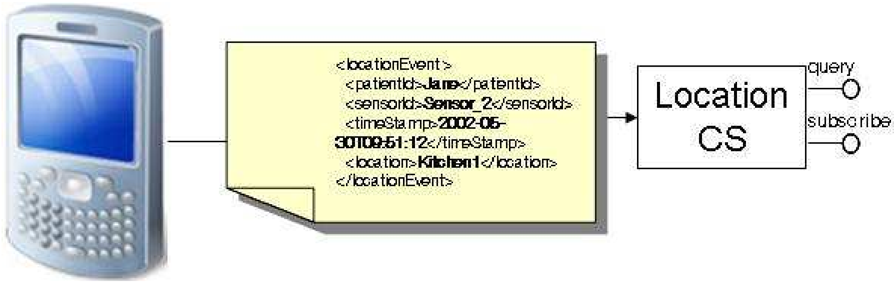


Fig. 1. Wrapping a sensor as a Context Source

point is that once this connection has been set, any access to the sensor is done through the CS component, as far as context management is concerned.

The job of "location CS" is to set semantic annotations to every bit of the sensor raw data, so that it can be automatically interpreted within the context management process later on. Figure 2 displays the result of such annotation.

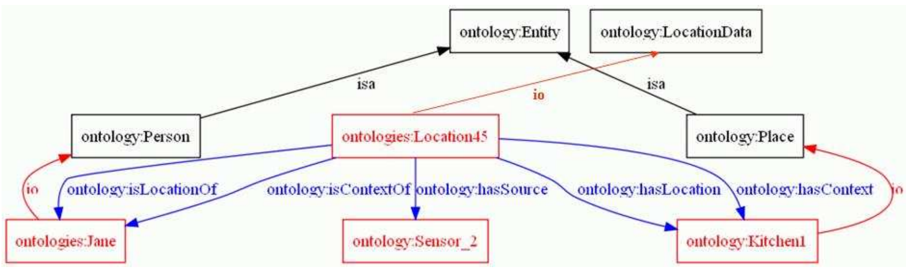


Fig. 2. Location context RDF model

For instance, "Kitchen1", which is the location value provided by the mobile terminal, has been interpreted as a "Place", which is a class in the context ontology. The annotation has been made explicit by linking the "Kitchen1" object to the "Place" class using a "io" ("instance of") relation. The result of this modeling process is presented in figure 2.

Once each sensor data has been modeled, aligning and aggregating them into a integrated and consistent model is straightforward, because they have been expressed along a common ontology. This consistent model is called a situation and is described in the next paragraph 3.1. The aggregation process is handled by the ContextStorage CC component. This component is introduced later on in paragraph 3.3.

3.1 Situation

As told previously, situations are built by aggregating context data. Situations model the states of the environment. A situation could be considered as a snapshot of the environment at a given point in time, which is made of whatever information about this environment we could collect from the sensors.

The algorithm we use for computing situations is inspired from the situation calculus introduced by McCarthy in 1963 [5]. The situation calculus is a logical formalism which makes it possible to reason over dynamical environments, and provide a solution to the question “what beliefs still holds in response to actions” [6]. With respect to our problem, a sensor event creates a transition from the current situation to the new situation, whenever the information it conveys is inconsistent with the current situation (e.g. the event reports that a light is on, while it is described as off in the current situation). In this case, a new situation is created which updates the current situation by adding the new information and removing the inconsistent part.

This process is carried out by the ContextStorage CS component, so that situations can be stored persistently once they have been created.

3.2 Similarity and clustering algorithms

The next goal of the LearningComponent CC is to proceed with a classification of the situations which have been stored over time as explained in the previous section. This classification process involves a similarity function and a clustering algorithm.

A similarity function allows to measure the similarity between two situations. It helps to differentiate two situations which are quite different or to assess the similarity of two situations which are close to each other. This function is a cornerstone of the classification process. As the items we would like to measure the similarity of are graphs, we have used two discrimination criteria:

1. concepts (nodes) that appear in the graph and how often they appear
2. relations between concepts of the graph

The first criteria is evaluated using the TF-IDF (for Term Frequency-Inverse Document Frequency) method [7]. This method has been originally introduced for text data mining, but we have adapted it to our problem by drawing a parallel between texts and situation graphs.

For the second criteria we have used Rada et al. [8] similarity measurement dedicated to semantic networks. This measurement is based on “is-a” hierarchical relations. Thus, in order to evaluate the similarity between two concepts in a model the shortest path between the two concepts in the “is-a” lattice is computed. This measure is applied node per node when comparing two graphs then results are added up and normalized.

Once normalized, these two measurements have been combined using a simple weighted sum.

Clustering aims at partitioning situations into groups of situations which are similar to each other. These groups are called clusters. If several situations occurring over time are very similar to each other, they will be grouped in the same cluster.

Thus large clusters will suggest recurring patterns among situations (contexts). In order to produce such clusters we have used the Markov Clustering algorithm (MCL). MCL [9] builds a NxN distance matrix where N is the number of elements (situations) and each matrix cell contains the distance between the column element and the line element. The algorithm then proceeds by simulating random walks within the distance matrix, by alternation of expansion and inflation stages. Expansion corresponds to computing random walks of higher length (with many steps). Inflation has the effect of boosting the probabilities of intra-cluster walks and will demote inter-cluster walks.

Iterating expansion and inflation results in the separation of the graph into different segments that we call clusters in our terminology. As mentioned previously in section 2, we expect clusters to correspond to relevant contexts. Each context would then be an abstraction of all the situations contained in its cluster.

3.3 architecture

The concepts introduced previously have been implemented and integrated within a prototype, which architecture is depicted in figure 3.

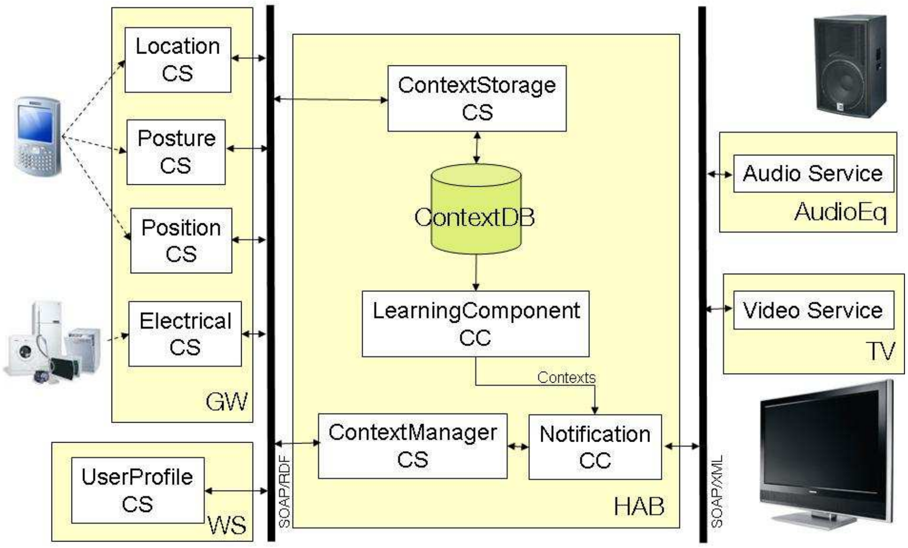


Fig. 3. Context Learning System Architecture

We simply recall and summarize the function of each component in the following:

Sensor Context Source : Provides a high level interface to sensors. A context source component can be viewed as a wrapper of the physical sensor.

Context Manager Context Source : This component subscribe to the different sensor context sources available. It integrates heterogeneous and disparate data conveyed by the Sensor Context Source events in order to build and maintain a consistent model of the world. Such a model is called a situation. In a previous paragraph 3.1, we explained how situations are built from sensor data events.

Notification Context Consumer : Analyses the world model, identifies critical situations, plans and triggers appropriate actions

Audio and video service : Render visual and audio information

Context Storage Context Source : Collects sensor data formats them into the context data description and stores them persistently. For more details the reader could refer to [10].

Learning Component Context Consumer : Analyses the situations stored over time, discovers and extracts recurring situations (contexts)

Context Data Storing : Collects sensor data formats into the context data description and stores them persistently for retrieval and postmortem and offline analysis.

After this short introduction of our approach and the description of our context learning prototype, we present the results obtained when applying our prototype to the data generated by the illustrative scenario exposed in section 2.

4 Experimental results

Enacting the scenario introduced in section 2 yields 31 sensor data events. These events are presented in figure 4. Each column of the table represent a value of a sensor measurement. Column values are grouped per sensor. For example the first column represents the switching on of the oven whereas the second one represents its switching off. Each line of the table corresponds to an event a sensor emits. Event lines are added in a chronological order, the first event (corresponding to “oven has been switched off”) is positioned as the first line of the table. For example, event number 14 is posted by the kitchen light, which reports the switching off of the light.

Events have been also plotted on the map, at the position Jane had when they occured. For example in figure 5-(b), we have plotted the events 5 to 13 events as circle shaped tags annotated with the number of the event. For instance, event 12 has been posted by the oven while it was switched on, whereas event 13 corresponding to its switching off.

Theses events have produced 27 situations, as resulting from the algorithm described in paragraph 3.1. Similarly to what we have done for the events, each situation has been plotted on the flat map between the couple of events that respectively initiated and terminated the situation. The 27 situations are then represented in figure 5-(c) as square shaped tags.

Although we model situations as RDF graphs, as explained in section 3.1, it is also convenient to represent them more concisely in terms of sensors measures as shown in table 6. This representation will be more suitable for evaluating the results of the algorithms as we’ll address this point in section 5.

The context learning component has identified 8 situations clusters, using the combined TF-IDF and Rada et al. similarity measure and the MCL clustering algorithm as explained in paragraph 3.2. These clusters and the situations they contain are presented in table 7.

	Oven		kitchen light		Bedroom light		Location		Posture			
	on	off	on	off	on	off	kitchen	bedroor	running	standin	seating	lying
1		1										
2				2								
3						3						
4								4				
5												5
6					6							
7											7	
8										8		
9						9						
10							10					
11			11									
12	12											
13		13										
14				14								
15								15				
16					16							
17											17	
18						18						
19												19
20					20							
21											21	
22										22		
23						23						
24							24					
25			25									
26				26								
27								27				
28					28							
29											29	
30						30						
31												31

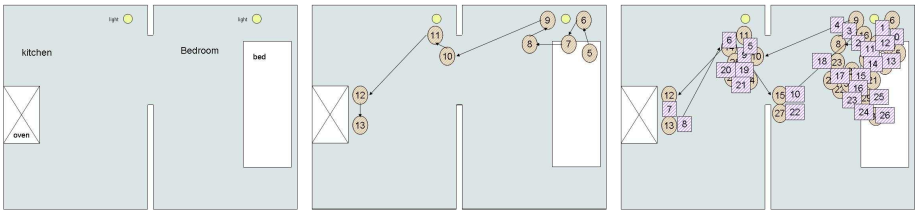
Fig. 4. Sensor events

For instance, cluster 0 contains the 4 situations 2, 12, 16, 24. If we check at their synthetic representation from table 7, we can notice that they are identical as shown in figure 8. Figure 8-(a) highlights the locations of Jane during the four situations 2, 12, 16, 24, while figure 8-(b) is an excerpt of table 7 corresponding to those situations.

We can notice that this cluster can be informally described as: ”The person is seating on his/her bed, while the light is on”.

With a similar analysis for all the clusters found we come out with the following interpretation:

Cluster 0 : ”The person is seating on his/her bed, while the light is on”
Cluster 1 : ” The person is standing in her/his bedroom, while the light is on ”
Cluster 2 : ” The person is standing in her/his bedroom, while the light is off ”
Cluster 3 : ” The person is standing in the kitchen, while the light is off ”
Cluster 4 : ” The person is standing in the kitchen, while the light is on ”
Cluster 5 : ” The person is in his/her bed, while the light is off ”
Cluster 6 : ” The person is lying on his/her bed, while the light is on”



(a) (b) (c)

Fig. 5. Environment, sensor events and situations

Cluster 7 : " The person is seating on his/her bed, while the light is off"

Now that we've exposed the results obtained using our approach, we would like to discuss them and position our work with respect to work done elsewhere in the next section.

5 Discussion

Before evaluating our experimental results, we would like to make a general comment on the relevancy of using sensors for observing and analyzing people behaviours in their ordinary daily life.

When installing our 5 sensors (oven, kitchen light, bedroom light, location sensor, posture sensor) in Jane's two rooms flat, as each of these sensors produces measurements within ranges of size 2 ('on'/'off' for the three first sensors, 'kitchen'/'bedroom' for the location sensor) and 4 ('running'/'standing'/'seating'/'lying' for the posture sensor) we could expect situations to span over more than 2 x 2 x 2 x 2 x 4 = 64 variants or potential combinations. However, although the scenario generates 27 situations, as seen on table 6, only few of them happen. We believe that this confirms the value of sensors, be they simple and sparsely deployed as in our experimental environment, for monitoring people behaviour. For instance, if we were to observe a concentration of situations which description fall outside those which usually happen, for example with the person lying while she/he is in the kitchen, we could consider it as an hint that something is going wrong.

Now back to our context learning research work, we can assert that our approach is able to identify clusters of similar situations which occur frequently. Although we haven't pushed the implementation of our approach that far yet, we could notice that some of these clusters correspond to contexts that are relevant to control the environment. For instance, cluster 1 and cluster 2 correspond to the context where the person is leaving her/his bedroom, and that their description suggest the bedroom light to be switched off (this is the only difference between the synthetic description of the two clusters).

Some work has addressed the extensive use of sensors measurements for learning human behaviour ([11]) but they have been limited in scope to the inference of user context (user activity/user task) from physical context information.

Fig. 6. Situations found

We think that these limitations principally stems from their use of the 'attribute/value' representation paradigm for representing context data. We believe that relations and structural information matter in context aware computing. For example, in a context aware building access control system, it makes sense to know the kind of relationship between the visitor and the people present in the building, and if there are several visitors it make sense to know the relationship between those visitors and to take this information into account when making a decision on which access policy to adopt.

In our approach we have used RDF which makes relational and structural information explicit, to model the instances of the population, we've learned reccurent context from. There are some existing learning techniques which are dedicated to structured data such as structural learning, multi-table learning, inductive logic programming (ILP).

Within a preliminary stage of our work we have evaluated and compared various clustering algorithms including the Kmean algorithm, the hierarchical classification and MCL. These methods are unsupervised classifiers, which basically means that no oracle is required to declare which class a sample belongs to. Kmean algorithm places each element of the population iteratively in one of K distinct classes which minimizes the its distance to the class. Each class is represented by a prototype (or centroïd) which is itself an element that represents the class. This prototype is updated at each iteration so as to ensure a good representation of the class. This iterative process completes as soon as an iteration doesn't change neither an element to class assignment, nor a prototype change in a class. There are two major drawbacks with the Kmean algorithm. One is

Clusters	Situations
C0	2 12 16 24
C1	3 11 17 23
C2	4 10 18 22
C3	5 9 19 21
C4	6 7 8 20
C5	0 14 26
C6	1 15
C7	13 25

Fig. 7. Clusters extracted

that K, the number of classes, has to be fixed arbitrarily, the other is that its results are very sensitive to the choice of the prototype at the boostraping stage.

We have evaluated another clustering algorithm called Hierarchical agglomerative clustering [12] that doesn't present the first drawback. This algorithm starts with singleton clusters where each element forms a cluster. The algorithm then proceeds by iteratively merging (agglomerating) pairs of clusters that are close to each other (in terms of similarity measure), until all clusters have been merged into a single cluster that contains the whole population. The result of this algorithm is a hierarchy of clusters, which can be represented as a dendogram. This algorithm shares the second drawback of the Kmeans algorithm because the number of clusters depends on the level at wich the dendogram is cut.

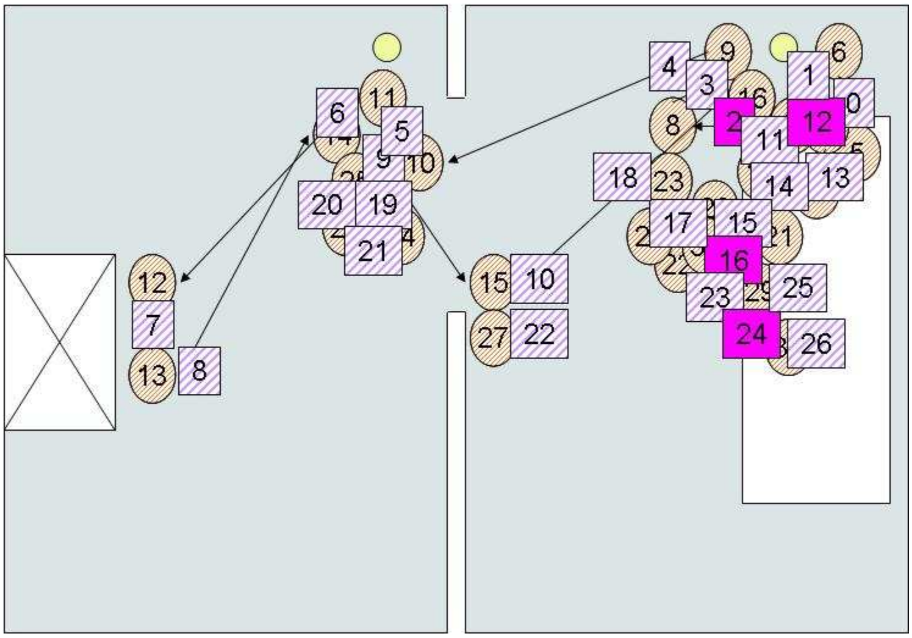
The MCL algorithm which we finally retained just ignores this second drawback. As we've seen, this algorithm had good performance on our scenario dataset.

The system has been assessed on several datasets, some of them involved a large amount of data. These experiments have revealed that some optimization in the data management and algorithm is required, if we need to increase the number of context sources, or if we need to store over a longer period of time, say several weeks. We now conclude and outline some perspectives of our work.

6 Conclusion and perspectives

In this paper, we have presented a system for archiving and mining data collected from sensors deployed in a home environment. The sensors we have used in our MIDAS project include white goods equipment and mobile terminal based sensors. From the data produced by these sensors we can retrieve the location, position and posture of their owners.

However, the flexibility of the data representation language we have adopted makes it possible to support a large variety of data sources, such as web services or personal



(a)

Oven		kitchen light		Bedroom light		Location		Posture				
on	off	on	off	on	off	kitchen	bedroom	running	standing	seating	lying	situations
												2
												12
												16
												24

(b)

Fig. 8. Position and description of situations in cluster 0

productivity tools (agenda, phonebook,...). From this archive we have applied data mining tools for extracting clusters of similar data. We have applied the system to a simple but realistic scenario of a person moving around in her flat. The method is able to detect recurring patterns. More over, all patterns found are relevant for automating the control of some devices. For instance, among the 8 patterns found, 4 of them describe a context where the light of the room the person is located in, should be switched off, whereas the other 4 describe a context where the light should be switched on.

Beyond context aware home automation, we believe that our approach is applicable to domains where similarity based clusters should be found out of structures of heterogeneous and disparated data. Hence the following application domains are potential targets of our system:

- Customer Relationship Management (Learn customers habits)
- Content search and casting (Learn customers preferences)

– SmartCity, SmartHome, SmartBuilding (Discover hidden correlations)
– Web services (context aware WS)

There are some issues remaining that we are currently addressing. They include scalability and the possibility to learn service context adaptation. For the second point, we expect machine learning mechanisms will allow the identification of correlation between service configuration parameters and context descriptions.

References

1. Lapkin, A.: Context-aware computing: A looming disruption. Research report, Gartner Inc. (2009)
2. Ramparany, F., Benazzouz, Y., Chotard, L., Coly, E.: Context aware assistant for the aging and dependent society. In et al., J.A., ed.: Workshop Proceedings of the 7th International Conference on Intelligence Environments, Nottingham, UK, University of Trent, IOS Press (2011) 798–809
3. Lefort, L., Henson, C., Taylor, K., Barnaghi, P., Compton, M., Corcho, O., Garcia-Castro, R., Graybeal, J., Herzog, A., Janowicz, K., Neuhaus, H., Nikolov, A., Page, K.: Semantic sensor network xg final report, W3C Incubator Group Report (2011) Available as `http://www.w3.org/2005/Incubator/ssn/XGR-ssn/`.
4. Ramparany, F., Poortinga, R., Stikic, M., Schmalenströer, J., Prante, T.: An open Context Information Management Infrastructure - the IST-Amigo Project. In of Engineering, I.I., Technology, eds.: Proceedings of the 3rd IET International Conference on Intelligent Environments (IE'07), Germany, University of Ulm (2007) 398–403
5. McCarthy, J.: Situations, actions and causal laws. Technical Report Memo2, Stanford Artificial Intelligence Project (1963)
6. McCarthy, J., Hayes, P.: Some philosophical problems from the standpoint of artificial intelligence. In Meltzer, B., Michie, D., eds.: Machine Intelligence. Volume 4. Edinburgh University Press (1969) 463–500
7. Salton, G., McGill, M.: Introduction to Modern Information Retrieval. McGraw-Hill International Editions (1983)
8. Rada, R., Mili, H., Bicknell, E., Blettner, M.: Development and application of a metric on semantic nets. IEEE Transactions on Systems, Man and Cybernetics **19** (1989) 17–30 ISSN 0018-9472.
9. van Dongen, S.: Graph Clustering by Flow Simulation. Phd thesis, University of Utrecht (2000)
10. Benazzouz, Y., Beaune, P., Ramparany, F., Boissier, O.: Modeling and storage of context data for service adaptation. In Sheng, Q.Z., Yu, J., Dustdar, S., eds.: Enabling Context-Aware Web Services: Methods, Architectures, and Technologies. Chapman and Hall/CRC (2010) 469–494
11. Brdiczka, O., Langet, M., Maisonnasse, J., Crowley, J.L.: Detecting human behavior models from multimodal observation in a smart home. IEEE transactions on automation sciences and engineering **6** (2009) 588–597
12. Day, W.H., Edelsbrunner, H.: Efficient algorithms for agglomerative hierarchical clustering methods. Journal of Classification **1** (1984) 1–24

Semantic Sensor Data Search in a Large-Scale Federated Sensor Network

Jean-Paul Calbimonte[1], Hoyoung Jeung[2], Oscar Corcho[1], and Karl Aberer[2]

[1]Ontology Engineering Group, Departamento de Inteligencia Artificial,
Facultad de Informática, Universidad Politécnica de Madrid, Spain
jp.calbimonte@upm.es,ocorcho@fi.upm.es
[2]School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland
hoyoung.jeung@epfl.ch,karl.aberer@epfl.ch

Abstract. Sensor network deployments are a primary source of massive amounts of data about the real world that surrounds us, measuring a wide range of physical properties in real time. However, in large-scale deployments it becomes hard to effectively exploit the data captured by the sensors, since there is no precise information about what devices are available and what properties they measure. Even when metadata is available, users need to know low-level details such as database schemas or names of properties that are specific to a device or platform. Therefore the task of coherently searching, correlating and combining sensor data becomes very challenging. We propose an ontology-based approach, that consists in exposing sensor observations in terms of ontologies enriched with semantic metadata, providing information such as: which sensor recorded what, where, when, and in which conditions. For this, we allow defining virtual semantic streams, whose ontological terms are related to the underlying sensor data schemas through declarative mappings, and can be queried in terms of a high level sensor network ontology.

1 Introduction

Sensors are related to a large number of human activities. They can be found in almost every modern monitoring system, including traffic management, health monitoring, safety services, military applications, environmental monitoring, and location-aware services. In such applications, sensors capture various properties of physical phenomena, hence becoming a major source of streaming data.

This growing use of sensors also increases the difficulty for applications to manage and query sensor data [1]. This difficulty becomes even more noticeable when applications need to search for a particular information set over federated and heterogeneous sensor networks, providing huge volumes of sensor data to large user communities [2]. In these environments, sensors from different vendors and with specific characteristics are installed and added to a system. Each of them produces different values, with different data schemas, precision or accuracy, and in different units of measurement. This heterogeneity complicates the task of querying sensor data as well as the corresponding metadata.

A rich body of research work has addressed the problem of querying data in large-scale sensor networks [3,4,5,6]. These studies generally focused on indexing sensor data, caching query results, and maximizing the shares of data to be carried together over networks. Whilst these methods substantially improve the query processing performance, they do not sufficiently consider the importance and difficulty of heterogeneous (sensor) data integration. In contrast, studies on semantic-aware sensor data management [7,8,9,10,11] have introduced a wide variety of mechanisms that search and reason over semantically enriched sensor data, while considering the heterogeneous characteristics of sensing environments. However, these proposals are still insufficient to show how to manage sensor data and metadata in a federated sensor network, and to efficiently process queries in a distributed environment.

This paper proposes a framework that enables efficient ontology-based querying of sensor data in a federated sensor network, going beyond state-of-the-art storage and querying technologies. The key features of the framework are briefly highlighted as follows:

– Our framework supports semantic-enriched query processing based on ontology information—for example, two users may name two sensors as of types "temperature" and "thermometer", yet the query processing in the framework can recognize that both sensors belong to the same type and include them in query results.
– The framework employs the SSN ontology[1], along with domain-specific ontologies, for effectively modeling the underlying heterogeneous sensor data sources, and establishes mappings between the current sensor data model and the SSN ontology observations using a declarative mapping language.
– The framework enables scalable search over distributed sensor data. Specifically, the query processor first looks up ontology-enabled metadata to effectively find which distributed nodes maintain the sensor data satisfying a given query condition. It then dynamically composes URL API requests to the corresponding data sources at the distributed GSN[2] nodes.
– Our framework has been developed in close collaboration with expert users from environmental science and engineering, and thus reflects central and immediate requirements on the use of federated sensor networks of the affected user community. The resulting system has been running as the backbone of the Swiss Experiment platform[3], a large-scale real federated sensor network.

The paper is organized as follows: we first describe in Section 2 the process of modeling metadata using the SSN ontology, and discuss the mappings between sensor data and the SSN observation model. In Section 3 we introduce the ontology-based query translation approach used in our framework. Section 4 describes the system architecture and its components, and in Section 5 we provide details about technical experimentations of our approach.We then discuss about relevant related work in Section 6, followed by our conclusions in Section 7.

[1] W3C Semantic Sensor Network (SSN-XG) Ontology [12]
[2] Global Sensor Networks [13], streaming data middleware used for the prototype.
[3] Swiss-Experiment: `http://www.swiss-experiment.ch/`

2 Modeling Sensor Data with Ontologies

Ontologies provide a formal, usable and extensible model that is suitable for representing information, in our case sensor data, at different levels of abstraction and with rich semantic descriptions that can be used for searching and reasoning [1]. Moreover in a highly heterogeneous setting, using standards and widely adopted vocabularies facilitates the tasks of publishing, searching and sharing the data.

Ontologies have been used successfully to model the knowledge of a vast number of domains, including sensors and observations [14]. Several sensor ontologies have been proposed in the past (see Section 6), some of them focused on sensor descriptions, and others in observations [14]. Most of these proposals are, however, often specific to a project, or discontinued, which do not cover many important areas of the sensor and observation domain. Moreover many of these ontologies did not follow a solid modeling process or did not reuse existing standards. In order to overcome these issues the W3C SSN XG group [12] introduced a generic and domain independent model, the SSN ontology, compatible with the OGC[4] standards at the sensor and observation levels.

The SSN ontology (See Fig. 1) can be viewed and used for capturing various properties of entities in the real world. For instance it can be used to describe sensors, how they function and process the external stimuli. Alternatively it can be centered on the observed data, and its associated metadata [15]. In this study, we employ the latter ontology modeling approach in a large-scale real sensor network application, the Swiss Experiment. For instance consider a wind-monitor sensor in a weather station deployed at a field site. The sensor is capable of measuring the wind speed on its specific location. Suppose that another sensor attached at the same station reports air temperature every 10 minutes. In terms of the SSN ontology both the wind and temperature measurements can be seen as observations, each of them with a different feature of interest (wind and air), and each referring to a different property (speed and temperature).

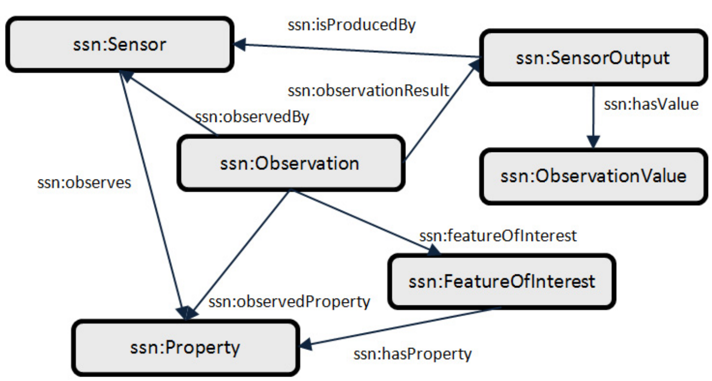


Fig. 1. Main concepts of the SSN ontology.

[4] Open Geospatial Consortium: `http://www.opengeospatial.org/`

In the SSN ontology, instances of the `Observation` class represent such observations, e.g. Listing 1.1, and are linked to a certain feature instance through a `featureOfInterest` property. Similarly the `observedProperty` links to an instance of a property, such as speed. Since the SSN model is intended to be generic, it does not define the possible types of observed properties, but these can be taken from a specialized vocabulary such as the NASA SWEET[5] ontology. Actual values of the sensor output can also be represented as instances linked to the `SensorOutput` class through the `hasValue` property. The data itself can be linked through a specialized property of a quantity ontology (e.g. the QUDT[6] `numericValue` property). Finally the observation can be linked to a particular sensor (e.g. Sensor instance `SensorWind1` through the `observedBy` property). Evidently more information about the observation can be recored, including units, accuracy, noise, failures, etc. Notice that the process of ontology modeling requires reuse and combination of the SSN ontology and domain-specific ontologies.

```
swissex:WindSpeedObservation1 rdf:type ssn:Observation;
      ssn:featureOfInterest [ rdf:type sweet:Wind];
      ssn:observedProperty  [ rdf:type sweetProp:Speed].
      ssn:observationResult
         [ rdf:type ssn:SensorOutput;
           ssn:hasValue [qudt:numericValue "6.245"^^xsd:double]];
      ssn:observedBy swissex:SensorWind1;
```

Listing 1.1. Wind Speed observation in RDF according to the SSN ontology

In our framework, we also model the sensor metadata. For example we can specify that the weather station platform where both sensors are installed, is geospatially located, using the SG84 vocabulary[7]. In the example in Listing 1.2, the location (latitude and longitude) of the platform of the `SensorWind1` sensor is provided. We can also include other information such as a responsible person, initial date of the deployment, etc.

```
swissex:SensorWind1 rdf:type ssn:Sensor;
   ssn:onPlatform [:hasGeometry [rdf:type wgs84:Point;
                                 wgs84:lat "46.8037166";
                                 wgs84:long "9.7780305"]];
   ssn:observes [rdf:type sweetProp:WindSpeed] .
```

Listing 1.2. Representation of a Sensor on a platform and its location in RDF

Although the observation model provides a semantically enriched representation of the data, sensors generally produce streams of raw data with very little structure and thus there is a gap between the observation model and the original data. For instance both sensors in Listing 1.3 (`wan7` and `imis_wfbe`) capture wind speed measurements but have different schemas, each one stores the observed value in a different attribute. To query wind speed observations in these

[5] `http://sweet.jpl.nasa.gov/` NASA SWEET Ontology
[6] Quantities, Units, Dimensions and Data Types ontologies, `http://www.qudt.org/`
[7] Basic Geo WGS84 Votcabulary: `http://www.w3.org/2003/01/geo/`

settings, the user needs to know the names of the sensors, and the names of all different attributes that match with the semantic concept of wind speed. This is an error-prone task and is unfeasible when the number of sensors is large.

```
wan7: {wind_speed_scalar_av FLOAT, timed DATETIME}
imis_wbfe: {vw FLOAT, timed DATETIME}
```

Listing 1.3. Heterogeneous sensor schemas

We take an ontology mapping-based approach to overcome this problem. Although in previous works [16,17] sensor observations are provided and published as RDF and linked data, they do not provide the means and representation that allows querying live sensor data in terms of an ontological model. Going beyond these approaches, we propose using declarative mappings that express how to construct SSN Observations from raw sensor schemas, and for this purpose we use the W3C RDB2RDF Group, R2RML language[8] to represent the mappings. For example we can specify that for every tuple of the `wan7` sensor, an instance of a SSN `ObservationValue` must be created, using the mapping definition `Wan7WindMap` depicted in Fig. 2 (See Listing 1.4 for its R2RML representation).

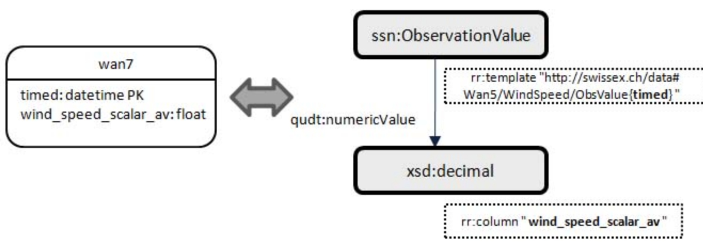


Fig. 2. Simple mapping from the `wan7` sensor to a SSN `ObservationValue`

The instance URI is composed according to the mapping `rr:template` rule that concatenates the `timed` column value to a prefix. The observation actual value is extracted from the `wind_speed_scalar_av` sensor field and is linked to the `ObservationValue` through a `qudt:numericValue` property.

```
:Wan7WindMap a rr:TriplesMapClass;
   rr:tableName "wan7";
   rr:subjectMap
     [rr:template
        "http://swissex.ch/data#Wan5/WindSpeed/ObsValue{timed}";
      rr:column "timed";
      rr:class ssn:ObservationValue;
      rr:graph swissex:WannengratWindSpeed.srdf ];
   rr:predicateObjectMap
     [ rr:predicateMap [ rr:predicate qudt:numericValue ];
       rr:objectMap    [ rr:column "wind_speed_scalar_av" ] ]; .
```

Listing 1.4. Mapping a sensor to a SSN ObservationValue in R2RML

[8] R2RML mapping language, `http://www.w3.org/2001/sw/rdb2rdf/r2rml/`

By using the mappings and the SSN ontology, we are able to express the sensor metadata and observations data using a semantic model, even if the underlying data sources are relational streams. In the next section we provide details about the query translation process that is carried out to make querying possible.

3 Querying Ontology-based Sensor Data

Ontology-based streaming data access aims at generating semantic web content from existing streaming data sources [18]. Although previous efforts have been made in order to provide semantic content automatically form relational databases using mappings [19], only recently this idea has been explored in the context of data stream management [18]. Our approach in this paper (Fig. 3) covers this gap, extending the work of [18] to support the R2RML syntax and produce algebra expressions that can be transformed into requests to federated sensor networks.

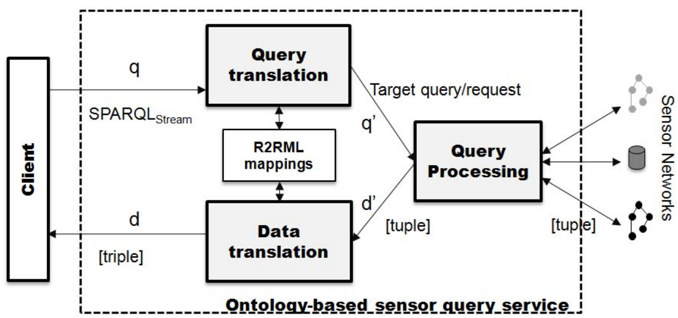


Fig. 3. Ontology-based sensor query service: translation of SPARQL$_{\text{Stream}}$ queries over virtual RDF streams, to requests over federated sensor networks

Our ontology-based sensor query service receives queries specified in terms of the SSN ontology using SPARQL$_{\text{Stream}}$ [18], an extension of SPARQL that supports operators over RDF streams such as time windows, and has been inspired by C-SPARQL [8]. Since the SPARQL$_{\text{Stream}}$ query is expressed in terms of the ontology, it has to be transformed into queries in terms of the data sources, using a set of mappings, expressed in R2RML. The language is used to define declarative mappings from relational sources to datasets in RDF, as detailed in Section 2. These are in fact *virtual* RDF *streams*, since they are not materialized beforehand, but the data is queried and transformed on demand after the SPARQL$_{\text{Stream}}$ query is translated. The target of this *query translation* process is a streaming query expression over the sensor streams. These queries are represented as algebra expressions extended with time window constructs, so that optimizations can be performed over them and can be easily translated to a target language or stream request, such as an API URL, as we will see in Section 4.

As an example, consider the mapping in Fig. 4, which extends the one displayed before in Fig. 2. This mapping generates not only the `ObservationValue` instance but also a `SensorOutput` and an `Observation` for each record of the sensor `wan7`. Notice that each of these instances constructs its URI with a different template rule and the `Observation` has a `observedProperty` property to the `WindSpeed` property defined in the SWEET ontology.

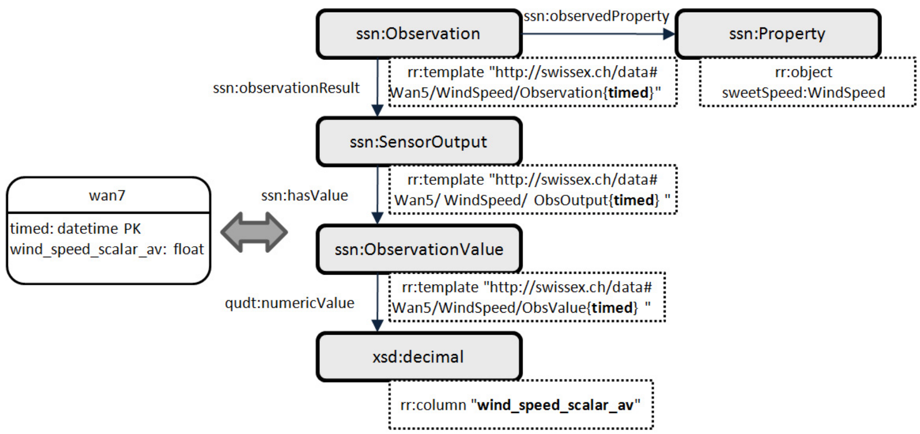


Fig. 4. Mapping from the `wan7` sensor to a `Observation` and its properties

The following query (Listing 1.5), obtains all wind-speed observation values greater than some threshold (e.g. 10) in the last 5 hours, from the sensors virtual RDF stream `swissex:WannengratWindSensors.srdf`. Such queries are issued by geo-scientists to collect filtered observations and feed their prediction models.

```
PREFIX ssn: <http://purl.oclc.org/NET/ssnx/ssn#>
PREFIX swissex: <http://swiss-experiment.ch/metadata#>
PREFIX qudt: <http://data.nasa.gov/qudt/owl/qudt#>
PREFIX sweetSpeed: <http://sweet.jpl.nasa.gov/2.1/propSpeed.owl#>
SELECT ?speed ?obs
FROM NAMED STREAM swissex:WannengratWindSpeed.srdf [NOW - 5 HOUR ]
WHERE {
  ?obs       a ssn:Observation;
             ssn:observationResult ?result;
             ssn:observedProperty ?prop.
  ?prop      a sweetSpeed:WindSpeed.
  ?result    ssn:hasValue ?obsvalue.
  ?obsvalue  a ssn:ObservationValue;
             qudt:numericValue ?speed.
  FILTER ( ?speed > 10 )  }
```

Listing 1.5. SPARQL$_{\mathrm{Stream}}$ query

Using the mapping definitions, the query translator can compose the corresponding algebra expression that creates a time window of 5 hours over the `wan7` sensor, applies a selection with the predicate `wind_speed_scalar_av > 10`, and finally projects the `wind_speed_scalar_av` and `timed` columns (See Fig. 5).

The algebra expressions can be transformed to continuous queries in languages such as CQL [20] or SNEEql [21], and then executed by a streaming query engine. In the case of GSN as the query engine, the algebra expression can be used to produce a sensor data request to the stream query engine. Specifically,

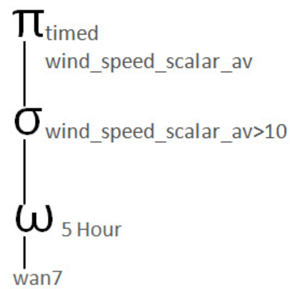


Fig. 5. Translation of the query in Listing 1.5 to an algebra expression, using the R2RML mappings.

the query engine in our framework processes the requests and returns a result set that matches the SPARQL$_{\text{Stream}}$ criteria. To complete the query processing, the result set is transformed by the *data translation* process to ontology instances (SPARQL bound variables or RDF, depending if it is a SELECT or a CONSTRUCT query).

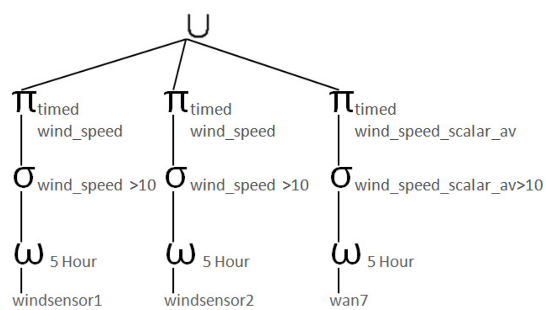


Fig. 6. Algebra UNION expression, with two additional wind-speed sensors.

Depending on the mappings available, the resulting algebra expression can become entirely different. For instance, suppose that there are similar mappings for the `windsensor1` and `windsensor2` sensors, also measuring wind-speed values as `wan7`. Then the resulting expression would be similar to the one in Fig. 6, but including all three sensors in a UNION expression. Conversely, a mapping for a sensor that observes a property different than `sweetSpeed:WindSpeed` will be ignored in the translation process for the sample query.

4 System Overview

Using the ontology-based approach for streaming data described in the previous section, we have built a sensor data search prototype implementation for the Swiss-Experiment project. The system (Fig. 7) consists of the following main components: the user interface, the federated GSN stream server instances, the sensor metadata repository and the ontology-based sensor query processor.

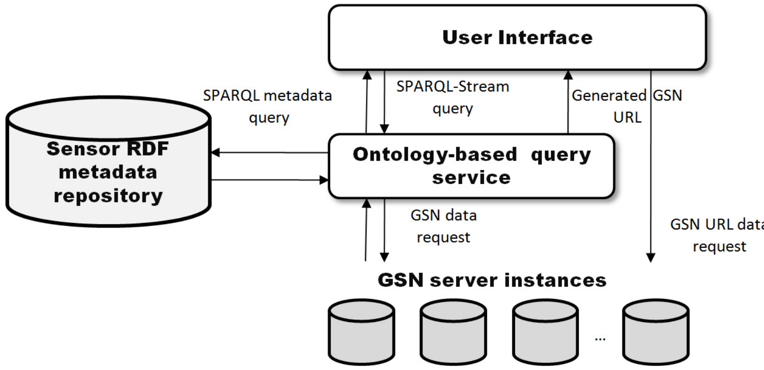


Fig. 7. System architecture

4.1 User Interface

The web-based user interface is designed to help the user filtering criteria to narrow the number of sensors to be queried (Fig. 8). Filtering criteria may include the sensing capabilities of the devices, e.g. select only the sensors that measure air temperature or wind speed. It is also possible to filter according to the characteristics of the deployment or platform, e.g. select sensors deployed in a particular region, delimited by a geo-referenced bounding box. It is also possible to filter by both data and metadata parameters. For instance the user may filter only those sensors registering air temperature values higher than 30 degrees. The filtering parameters can be passed to the ontology-based query processor, as a SPARQL$_{\text{Stream}}$ query in terms of the SSN ontology as detailed next.

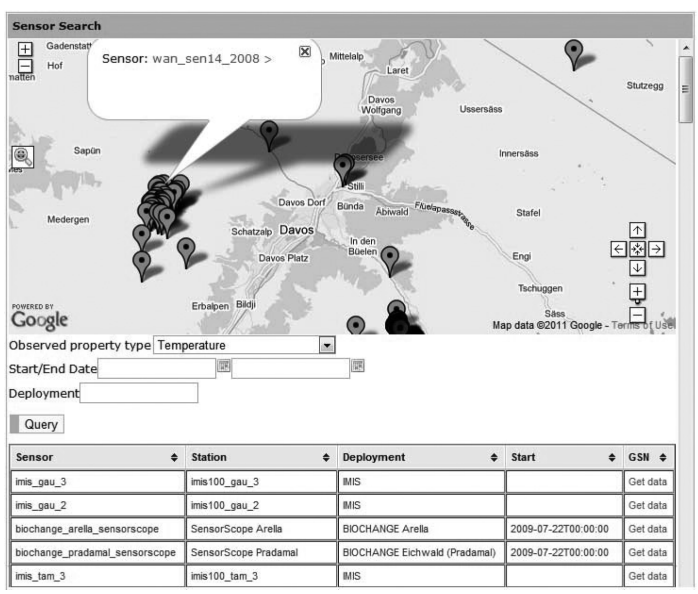


Fig. 8. Sensor data search user interface

4.2 Ontology-based Sensor Query Processor

This component is capable of processing the SPARQL$_{\text{Stream}}$ queries received from the user interface, and perform the query processing over the metadata repository and the GSN stream data engine. The ontology-based processor uses the previously defined R2RML mappings and the sensor metadata in the RDF repository to generate the corresponding requests for GSN, as explained in Section 3.

The ontology-based query service delegates the processing to the GSN server instances by composing *data requests* according to the GSN web-service or URL interfaces. In the case of the web service, a special GSN wrapper for the WSDL specification[9] has been developed, that can be used if the user requires to obtain the observations as RDF instances, just as described in Section 3. Alternatively, the ontology-based sensor query processor can generate GSN API[10] URLs from the algebra expressions. These URLs link directly to the GSN server that provides the data with options such as bulk download, CSV formatting, etc.

```
http://montblanc.slf.ch:22001/multidata?vs[0]=wan7&
      field[0]=wind_speed_scalar_av&
      from=15/05/2011+05:00:00&to=15/05/2011+10:00:00&
      c_vs[0]=wan7s&c_field[0]=wind_speed_scalar_av&c_min[0]=10
```

Listing 1.6. Generation of a GSN API URL

For example, the expression in Fig. 5 produces the GSN API URL in Listing 1.6. The first part is the GSN host (`http://montblanc.slf.ch:22001`). Then the sensor name and fields are specified with the `vs` and `field` parameters. The `from-to` part represents the time window and finally the last line specifies the selection of values greater than 10 (with the `c_min` parameter). These URLs are presented in each sensor info-box in the user interface map.

With this semantically enabled sensor data infrastructure, users can issue complex queries that exploit the existing relationships of the metadata and also the mappings, such as the one in (Listing 1.7).

```
PREFIX ssn: <http://purl.oclc.org/NET/ssnx/ssn#>
PREFIX omgeo: <http://www.ontotext.com/owlim/geo#>
PREFIX dul: <http://www.loa-cnr.it/ontologies/DUL.owl#>
PREFIX swissex: <http://swiss-experiment.ch/metadata#>
PREFIX sweet: <http://sweet.jpl.nasa.gov/2.1/prop.owl#>
SELECT ?obs ?sensor
FROM NAMED STREAM swissex:WannengratSensors.srdf [NOW - 5 HOUR ]
WHERE {
  ?obs        a ssn:Observation;
              ssn:observedBy ?sensor.
  ?sensor     ssn:observes ?prop;
              ssn:onPlatform ?platform.
  ?platform   dul:hasLocation [swissex:hasGeometry ?geo].
  ?geo        omgeo:within(46.85 9.75 47.31 10.08) .
  ?prop       a sweet:MotionProperty.     }
```

Listing 1.7. SPARQL$_{\text{Stream}}$ query for the ontology-based sensor metadata search

[9] GSN Web Service Interface: `http://gsn.svn.sourceforge.net/viewvc/gsn/branches/documentations/misc/gsn-webservice-api.pdf`

[10] GSN Web URL API: `http://sourceforge.net/apps/trac/gsn/wiki/web-interfacev1-server`

This query requests the observations and originating sensor in the last 5 hours, for the region specified by a bounding box, and only for those sensors that measure motion properties. The geo-location query boundaries are specified using the `omgeo:within` function, and RDF semantic stores such as OWLIM [11] use semantic spatial indexes to compute these kind of queries. Regarding the observed property, considering that the `MotionProperty` is defined in the SWEET ontology as a superclass of all motion-related properties such as Wind Speed, Acceleration or Velocity, all sensors that capture these properties are considered in the query.

In all these examples, the users do not need to know the particular names of the real sensors, nor they need to know all the sensor attribute names that represent an observable property. This clearly eases the task for a research scientist, who can easily use and access the data he needs, with little knowledge of the technical details of the heterogeneous sensor schemas and their definitions. Also, this framework enables easily plugging new sensors to the system, without changing any existing query and without programming. All previous queries would seamlessly include new sensors, if their metadata and mappings are present in the repository.

4.3 GSN Server Instances

Our ontology-based approach for sensor querying relies on the existence of efficient stream query engines that support live sensor querying and that can be deployed in a federated environment. In the Swiss-Experiment project, the sensor data is maintained with Global Sensor Networks (GSN)[13], a processor that supports flexible integration of sensor networks and sensor data, provides distributed querying and filtering, as well as dynamic adaptation and configuration.

The Swiss-Experiment project has several GSN instances deployed in different locations which operate independently. In this way they can efficiently perform their query operations locally, and can be accessed using the interfaces mentioned earlier. However the metadata for these instances is centralized in the RDF metadata repository, enabling the federation of these GSN instances as described in the previous subsection.

4.4 Sensor Metadata Repository

We have used the Sesame [12] RDF store for managing the centralized sensor metadata, using the SSN ontology.The entire set of sensor metadata is managed with the Sensor Metadata Repository (SMR)[2]. The SMR is a web-based collaborative environment based on Semantic Wiki technologies [22], which includes not only static metadata but also dynamic metadata including the information of outliers and anomalies or remarks on particular value sets. This system provides

[11] OWLIM: `http://www.ontotext.com/owlim`
[12] Sesame: `http://www.openrdf.org/`

an easy and intuitive way of submitting and editing their metadata without any programming.

In SMR each sensor, platform or deployment has an associated Wiki page where the data can be semantically annotated with attribute-value pairs, and entities can be connected to each other with semantic properties. This allows interlinking related pages and also dynamically generating rich content for the users, based on the annotated metadata. The entire contents of the SMR can be queried programmatically using the SPARQL language, making it usable not only for humans but also for machines.

5 Experimentation

In order to validate our approach we have conducted a series of experiments in the sensor data and metadata system described previously. The goals were to (i) analyze empirically the scalability of semantic sensor metadata queries and (ii) assess the query and data transformation overhead of our approach. For the first objective, we compared a straightforward (but currently used by scientists) way of obtaining all sensors that measure a particular property (e.g. temperature), with our approach. The former consists in getting sensor details form every sensor in every deployment in the distributed system, and then comparing the sensor attribute name with the property name.

In our environment we have 28 deployments (aprox. 50 sensors in each one), running on its own GSN instance accessible through a web service interface. Therefore to perform this operation the client must contact all of these services to get the required information, making it very inefficient as the number of deployments increases (See Fig. 9). Conversely, using our centralized semantic search we eliminated the need of contacting the GSN instances at all for this type of query, as it can be solved by exploring the sensor metadata, looking for those sensors that have a `ssn:observes` relationship with the desired property.

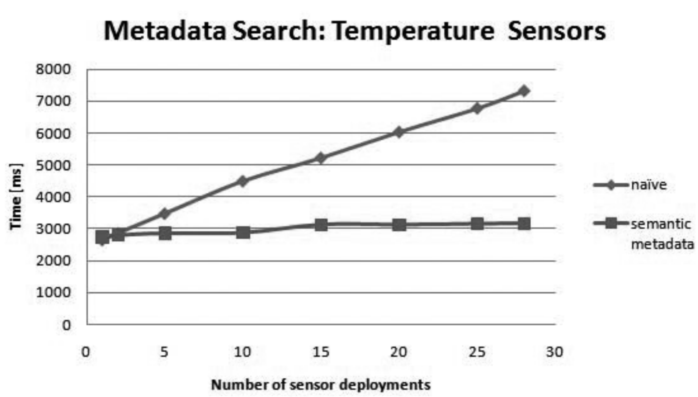


Fig. 9. Comparing metadata search: obtain all sensors that measure temperature. The naïve vs. semantic centralized approach.

As we see in Fig. 9 it is not only scalable as we add more deployments, but we also provide an answer that is independent of the syntactic name assigned to the sensor attributes.

Our approach sometimes incurs in a computing overhead when translating the SPARQL$_{\text{Stream}}$ queries to the internal algebra and the target language or URL request, using the mapping definitions. We analyzed this by comparing the query times of a raw GSN service request and a SPARQL$_{\text{Stream}}$ query translated to an equivalent GSN request. We executed this test over a single simulated deployment, first with only one sensor and up to 9 sensors with data updates every 500 ms. The query continuously obtains observations from the sensors in the last 10 minutes, filtering values smaller than a fixed constant, similarly to Listing 1.5.

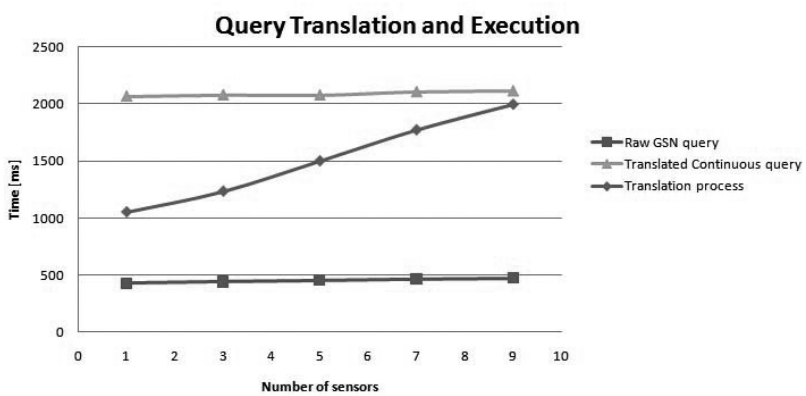


Fig. 10. Query execution and translation overhead: comparing a raw query vs. query translation.

As we show in Fig. 10 the overhead is of roughly 1.5 seconds for the test case. Notice that the overhead is seemingly constant as we add more sensors to the mappings. However this is a continuous query and the translation time penalty has been excluded form the computation, as this operation is only executed once, then the query can be periodically executed. In any case this additional overhead is also displayed in Fig. 10 and it degrades as the number of mappings to sensors increases. This is likely because mappings are stored and loaded as files, and not cached in any way. More efficient management of large collections of mappings could throw better results for the translation operation. Nevertheless we show that continuous queries have an acceptable overhead, almost constant for the chosen use-case.

6 Related Work

Several efforts in the past have addressed the task of representing sensor data and metadata using ontologies, and also providing semantic annotations and querying over these sources, as recounted below.

Ontology Modeling for Sensor Data The task of modeling sensor data and metadata with ontologies has been addressed by the semantic web research community in recent years. As recounted in [14], many of the early approaches focused only on sensor meta-information, overlooking observation descriptions, and also lacked the best practices of ontology reuse and alignment with standards. Recently, through the W3C SSN-XG group, the semantic web and sensor network communities have made an effort to provide a domain independent ontology, generic enough to adapt to different use-cases, and compatible with the OGC standards at the sensor level (SensorML[13]) and observation level (O&M[14]). These ontologies have also been used to define and specify complex events and actions that run on an event processing engine [23].

Semantic Sensor Queries and Annotations Approaches providing search and query frameworks that leverage semantic annotations and metadata, have been presented in several past works. The architectures described in [24] and [25], rely on bulk-import operations that transform the sensor data into an RDF representation that can be queried using SPARQL in memory, lacking scalability and the real-time querying capabilities.

In [10] the authors describe preliminary work about annotating sensor data with Linked Data, using rules to deduce new knowledge, although no details about the RDF transformation are provided. Semantic annotations are also considered for the specific task of adding new sensors to observation services in [9]. The paper points out the challenges of dynamically registering sensors, including grounding features to defined entities, to temporal, spatial context. In [2], the authors describe a metadata management framework based on Semantic Wiki technology to store distributed sensor metadata. The metadata is available through SPARQL to external services, including the system's sensor data engine GSN, that uses this interface to compute distributed joins of data and metadata on its queries.

In [26] a semantic annotation and integration architecture for OGC-compliant sensor services is presented. The approach follows the OGC-sensor Web enablement initiative, and exploits semantic discovery of sensor services using annotations. In [11] a SOS service with semantic annotations on sensor data is defined. The approach consists in adding annotations, i.e. embed terminology form an ontology in the XML O&M and SensorML documents of OGC SWE, using either XLink or the SWE swe:definition attribute for that purpose. In a different approach, the framework presented in [27] provides sensor data readings annotated with metadata from the Linked Data Cloud. While in this work we addressed the

[13] OGC SensorML: `http://www.opengeospatial.org/standards/sensorml`
[14] Observations & Measurements: `http://www.opengeospatial.org/standards/om`

problems related to heterogeneity of the data schemas, it is also worth mentioning that Linked Data initiatives can be helpful for integrating data from different (local or remote) publishers, unlike our use case where all the observations were centralized through GSN.

7 Conclusions

We presented an ontology-based framework for querying sensor data, considering metadata and mappings to underlying data sources, in a federated sensor network environment. Our approach reuses the SSN ontology along with domain-specific ontologies for modeling the sensor metadata so that users can pose queries that exploit their semantic relationships, therefore they do not require any knowledge about sensor specific names or their attributes or schemas. Users can just issue a high-level query that will internally look for the appropriate and corresponding sensors and attributes, according to the query criteria.

For this purpose we perform a dynamic translation of $\text{SPARQL}_{\text{Stream}}$ queries into algebra expressions that can be used to generate queries or data requests like the GSN API URLs, while extending the use of the R2RML language specification for streaming sensor data. As a result we have enabled distributed processing of queries in a federated sensor network environment, through a centralized semantic sensor metadata processing service. This approach has been implemented in the Swiss-Experiment project, in collaboration with users form the environmental science community, and we have built a sensor search prototype powered by our framework. We are planning to expand this work in the future, to integrate this platform with external data sources that may provide additional information about the sensors, including location, features of interest or other metadata. Finally we are considering the integration with other sensor data sources running under other platforms, which may be relevant in the domain.

Acknowledgements Supported by the myBigData project (TIN2010-17060) funded by MICINN (Spanish Ministry of Science and Innovation), and the european projects PlanetData (FP7-257641) and SemSorGrid4Env (FP7-223913).

References

1. Corcho, O., García-Castro, R.: Five challenges for the Semantic Sensor Web. Semantic Web **1**(1) (2010) 121–125
2. Jeung, H., Sarni, S., Paparrizos, I., Sathe, S., Aberer, K., Dawes, N., Papaioannou, T., Lehning, M.: Effective Metadata Management in Federated Sensor Networks. In: SUTC, IEEE (2010) 107–114
3. Motwani, R., Widom, J., Arasu, A., Babcock, B., Babu, S., Datar, M., Manku, G., Olston, C., Rosenstein, J., Varma, R.: Query processing, resource management, and approximation in a data stream management system. In: CIDR. (2003) 245–256
4. Ahmad, Y., Nath, S.: COLR-Tree: Communication-efficient spatio-temporal indexing for a sensor data web portal. In: ICDE. (2008) 784–793

5. Li, J., Deshpande, A., Khuller, S.: Minimizing communication cost in distributed multi-query processing. In: ICDE. (2009) 772 –783
6. Wu, J., Zhou, Y., Aberer, K., Tan, K.L.: Towards integrated and efficient scientific sensor data processing: a database approach. In: EDBT. (2009) 922–933
7. Compton, M., Neuhaus, H., Taylor, K., Tran, K.: Reasoning about sensors and compositions. In: SSN. (2009)
8. Barbieri, D.F., Braga, D., Ceri, S., Della Valle, E., Grossniklaus, M.: C-SPARQL: SPARQL for continuous querying. In: WWW '09, ACM (2009) 1061–1062
9. Bröring, A., Janowicz, K., Stasch, C., Kuhn, W.: Semantic challenges for sensor plug and play. Web and Wireless Geographical Information Systems (2009) 72–86
10. Wei, W., Barnaghi, P.: Semantic annotation and reasoning for sensor data. In: Smart Sensing and Context. (2009) 66–76
11. Henson, C., Pschorr, J., Sheth, A., Thirunarayan, K.: SemSOS: Semantic Sensor Observation Service. In: CTS, IEEE Computer Society (2009) 44–53
12. Lefort, L., Henson, C., Taylor, K., Barnaghi, P., Compton, M., Corcho, O., Garcia-Castro, R., Graybeal, J., Herzog, A., Janowicz, K., Neuhaus, H., Nikolov, A., Page, K.: Semantic Sensor Network XG final report, available at `http://www.w3.org/2005/Incubator/ssn/XGR-ssn/`. Technical report, W3C Incubator Group (2011)
13. Aberer, K., Hauswirth, M., Salehi, A.: A middleware for fast and flexible sensor network deployment. In: VLDB, VLDB Endowment (2006) 1199–1202
14. Compton, M., Henson, C., Lefort, L., Neuhaus, H., Sheth, A.: A survey of the semantic specification of sensors. In: SSN. (2009) 17
15. Janowicz, K., Compton, M.: The Stimulus-Sensor-Observation Ontology Design Pattern and its Integration into the Semantic Sensor Network Ontology. In: SSN. (2010) 7–11
16. Patni, H., Henson, C., Sheth, A.: Linked sensor data. In: Collaborative Technologies and Systems (CTS), 2010 International Symposium on, IEEE (2010) 362–370
17. Barnaghi, P., Presser, M., Moessner, K.: Publishing Linked Sensor Data. In: SSN. (2010)
18. Calbimonte, J., Corcho, O., Gray, A.: Enabling ontology-based access to streaming data sources. In: ISWC. (2010) 96–111
19. Sahoo, S.S., Halb, W., Hellmann, S., Idehen, K., Jr, T.T., Auer, S., Sequeda, J., Ezzat, A.: A survey of current approaches for mapping of relational databases to RDF. W3C (January 2009)
20. Arasu, A., Babu, S., Widom, J.: The CQL continuous query language: semantic foundations and query execution. The VLDB Journal **15**(2) (June 2006) 121–142
21. Brenninkmeijer, C.Y., Galpin, I., Fernandes, A.A., Paton, N.W.: A semantics for a query language over sensors, streams and relations. In: BNCOD '08. (2008) 87–99
22. Völkel, M., Krötzsch, M., Vrandecic, D., Haller, H., Studer, R.: Semantic Wikipedia. In: WWW '06, ACM (2006) 585–594
23. Taylor, K., Leidinger, L.: Ontology-driven complex event processing in heterogeneous sensor networks. In: ESWC. (2011) 285–299
24. Lewis, M., Cameron, D., Xie, S., Arpinar, B.: ES3N: A semantic approach to data management in sensor networks. In: SSN. (2006)
25. Huang, V., Javed, M.: Semantic sensor information description and processing. In: SENSORCOMM, IEEE (2008) 456–461
26. Babitski, G., Bergweiler, S., Hoffmann, J., Schön, D., Stasch, C., Walkowski, A.: Ontology-based integration of sensor web services in disaster management. GeoSpatial Semantics (2009) 103–121
27. Le-Phuoc, D., Parreira, J., Hausenblas, M., Han, Y., Hauswirth, M.: Live linked open sensor database. In: I-Semantics, ACM (2010) 1–4

A semantic infrastructure for a Knowledge Driven Sensor Web

Deshendran Moodley[1] and Jules Raymond Tapamo[2]

[1]School of Computer Science, University of KwaZulu-Natal, Durban, South Africa

[2]School of Electrical, Electronic and Computer Engineering, University of KwaZulu-Natal, Durban, South Africa

Abstract. Sensor Web researchers are currently investigating middleware to aid in the dynamic discovery, integration and analysis of vast quantities of high quality, but distributed and heterogeneous earth observation data. Key challenges being investigated include dynamic data integration and analysis, service discovery and semantic interoperability. However, few efforts deal with the management of both knowledge and system dynamism. Two emerging technologies that have shown promise in dealing with these issues are ontologies and software agents. This paper introduces the idea and identifies key requirements for a Knowledge Driven Sensor Web and presents our efforts towards developing an associated semantic infrastructure within the Sensor Web Agent Platform.

Keywords: sensor web, ontologies, multi-agent systems, semantic middleware

1 Introduction

Advances in sensor technology and space science have resulted in the availability of vast quantities of high quality, but distributed and heterogeneous earth observation data. Sensor Web researchers are currently investigating middleware to facilitate the dynamic discovery, integration and analysis of this data with the vision of creating a global worldwide Sensor Web [33][6][9]. Key challenges being investigated include dynamic data discovery, integration and analysis, semantic interoperability, and sensor tasking. While it has been acknowledged that abstractions are required to bridge the gap between sensors and applications [6][9] and to provide support for the rapid deployment of end user applications [9], the most effective mechanism for modeling and managing the resultant deluge of software components remains an open issue. Two emerging technologies in Computer Science that have shown promise in dealing with these challenges are software agents and ontologies. Agent researchers propose the use of software agents as logical abstractions to model and manage software components in large scale, dynamic and open environments [17][34][35]. Software agents are autonomous software components that communicate at the knowledge level [13][17]. Many agent based architectures have been proposed for the Sensor Web [14][23][5][2]. However most approaches have limited support for the construction and evolution of the ontologies to support domain modeling, agent communication and reasoning, and to represent the algorithms, scientific theories and beliefs that are routinely applied to sensor data. In previous work we described an agent based architecture for the Sensor Web [21], i.e. the Sensor Web Agent Platform (SWAP), and proposed initial components for the semantic infrastructure [31]. In this paper we introduce the idea of a knowledge driven Sensor Web and describe a semantic infrastructure that supports both the specification and integration of scientific theories and system modeling. Additional details of the implementation of the ontologies and the reasoners can be found in [20].

The rest of the paper is organised as follows. In section 2 key requirements of a Knowledge Driven Sensor Web and its potential impact is described. Section 3 reviews related research. The SWAP semantic infrastructure is described in section 4 and in section 5 we conclude with a summary of key contributions and some avenues for future work.

2 A Knowledge Driven Sensor Web

A global Sensor Web must not only deal with issues around the provision, fusion and analysis of heterogeneous data. It must also support knowledge capture and use. Knowledge includes data processing and transformation algorithms, scientific theories and even subjective beliefs. To use this knowledge a mechanism must exist to dynamically apply knowledge to observations and to combine the results into meaningful information for end users. This capability to capture and apply knowledge will lead to a *Knowledge Driven Sensor Web (KDSW).*

A semantic infrastructure for a KDSW must include support for:

- Data and knowledge dynamism: a comprehensive but integrated conceptual modeling framework that includes support for not only modeling theme, time and space, but also uncertainty
- System and application dynamism: modeling of system entities, services, workflows, agents (system dynamism) and seamless movement between the conceptual model and the system model to support continuous application and service deployment

Potential benefits of a Knowledge Driven Sensor Web (KDSW) include [22]:

- Promoting the sharing and reuse of data, knowledge and services
- Facilitating human collaboration and scientific experimentation

- Reducing information overload and system complexity
- Managing both data, knowledge and system dynamism
- Increasing automation and machine intelligence

A Knowledge Driven Sensor Web can provide specific benefits to a wide range of users in the earth observation community. Decision makers can access, manage and visualise information provided by real time monitoring applications. Earth observation scientists can capture and share earth observation data and knowledge, and use the Sensor Web as a platform for experimentation, collaboration and knowledge discovery. Developers can easily design, develop and deploy dynamic Sensor Web services and end user applications.

3 Related work

A number of agent based Sensor Web approaches exist. These include the Internet-scale resource-intensive sensor network services (IrisNet) [14], Abacus [2], the agent based imagery and geospatial processing architecture (AIGA) [23], and the approach by Biswas et al. [5]. A summary of these approaches is given in [21]. Each approach proposes some form of layered architecture that provide abstractions to separate sensor agents from data analysis and filtering agents and aims to ease the modeling of agent based applications. While these approaches are promising for single or even groups of organizations building distributed agent based applications, except for the limited support provided in AIGA [23], no explicit support is provided for creating and managing ontologies that are required for agent communication and processing in an open Internet scale multi-agent system [13][34][35].

Ontologies are being widely investigated within the geospatial community to standardise, dynamically integrate and query complex earth observation data. Agarwal [1] summarises key advances in ontology research within the geospatial community. A more recent survey by Compton et. al. [8] describes the range and expressive power of twelve sensor ontologies. Despite these efforts there are still many outstanding challenges. The added temporal and spatial dimension associated with geospatial data requires additional representation support for modeling and formalising the domain [1][3]. One intuitive approach to model geospatial entities is to follow the human cognition system. Humans store knowledge in three separate cognitive subsystems within the mind [19]. The *what* system of knowledge operates by recognition, comparing evidence with a gradually accumulating store of known objects. The *where* system operates primarily by direct perception of scenes within the environment, picking up invariants from the rich flow of sensory information. The *when* system operates through the detection of change over time in both stored object and place knowledge, as well as sensory information. Separate ontological representations for space, time and theme have been proposed [26][31]. However, these approaches still lack support for representing the inherent uncertainty [3] associated with sensor data or for representing system entities. Even the widely used Web Ontology Language (OWL) [25] still lacks core support for representing time, space and uncertainty [30] and for representing system entities such as agents, services and processes.

4 The SWAP semantic infrastructure

Fig. 1 shows the different ontologies provided by SWAP. Ontologies are split into two levels, a conceptual level and a technical level. Conceptual ontologies are used for modeling and representing observations and theories about the physical world. Technical ontologies are used for modeling and representing the software entities (agents) that will host and process these observations and theories.

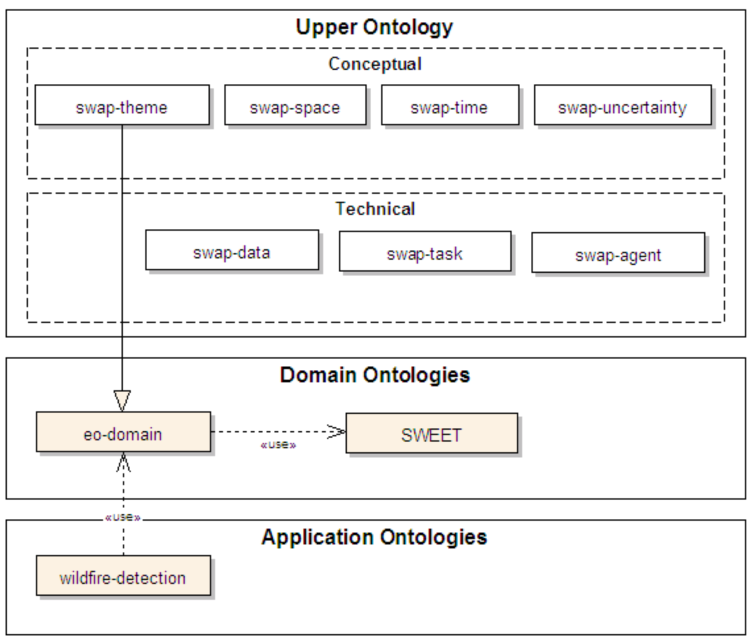


Fig. 1. SWAP ontology levels

The conceptual ontologies are based on creating separate subsystems as proposed by Mennis et al [19]. SWAP defines four conceptual dimensions to represent and reason about knowledge, the traditional dimensions of theme, space and time, and introduces a fourth dimension for uncertainty. An ontology and an associated reasoner is provided for each dimension. The reasoners currently use different inferencing engines: the thematic reasoner uses a Pellet reasoner; the temporal and spatial reasoners use a Jena rule-based engine; and the uncertainty reasoner uses a Bayesian inference engine. Domain ontologies for specific application domains are built by extending the *swap-theme* ontology. The *eo-domain* ontology extends the *swap-theme* ontology by adding concepts for building applications in the earth observation domain (Fig. 1). It currently references concepts from the SWEET [27] ontologies, an existing set of earth science ontologies. Application ontologies specify concepts that are used for specific applications, e.g. wildfire detection. Application specific concepts are specified along one or more of the four dimensions. The four reasoners are applied independently as required to perform inferencing on the application ontology.

4.1 The thematic dimension

The thematic dimension provides a thematic viewpoint for representing and reasoning about thematic concepts. The *swap-theme* ontology provides for the representation of observations and is based on the OGC's model of observations and measurements [10]. The *Observation* concept, defined in the *swap-theme* ontology, describes a single or a set of observations. Various thematic, spatial, temporal or uncertainty properties that are known may be specified for an observation (Fig. 2). The different types of properties are defined in the respective conceptual ontologies, e.g. thematic properties are defined in the *swap-theme* ontology and spatial properties are defined in the *swap-space* ontology.

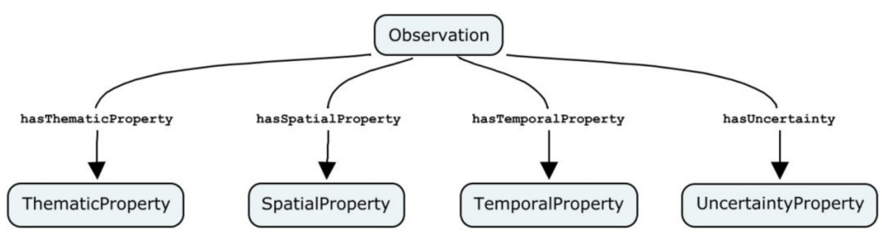


Fig. 2. Representing an observation

Two thematic properties are defined in *swap-theme*, *observesEntity* describes the entity being observed (*observedEntity*), while *observesProperty* describes the property of the entity that is being measured (*observedProperty*). The *eo-domain* ontology (Fig. 3) links observable properties from the NASA SWEET [27] property ontology by making these properties a subclass of *observedProperty* such as

BrightnessTemperature[1] and DryBulbTemperature[2]. Geographical entities from the SWEET earthrealm and SWEET phenomena ontologies are also linked by making these entities a subclass of *observedEntity*, e.g. *Air*, *Ocean*, *PlanetarySurface* and *Wind*.

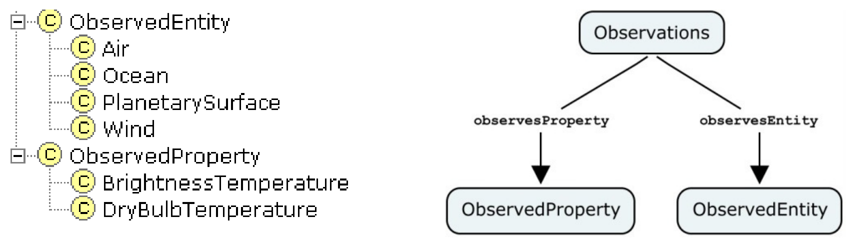


Fig. 3. The *eo-domain* ontology and representing a data set of observations

The schema for the thematic reasoner consists of the *eo-domain*, *swap-theme* and the SWEET ontology. This allows the inference engine to infer relations with SWEET concepts not explicitly referenced in the *eo-domain* ontology, e.g. that *BrightnessTemperature* and *DryBulbTemperature* are both subclasses of *Temperature*.

4.2 The spatial and temporal dimensions

The *swap-space* ontology provides concepts for representing and reasoning about the spatial aspects of data. A part of the *swap-space* ontology is shown in Fig. 4. Spatial entities include spatial reference systems, spatial projections, spatial resolution and location. Locations can be common descriptions such as a point coordinate or a bounding box, or well defined spatial geometries such as a point, line or polygon. A *SpatialThing* is defined as an entity that has a *Location* and the spatial reasoner determines how two *SpatialThings* are related. Since OWL does not provide native support for spatial representation, a set of spatial rules were formulated using the Jena[3] rule-based OWL reasoner to represent the eight spatial operators specified in the OpenGIS simple features for SQL [24].

---

[1] brightness temperature is the measure of the intensity of radiation thermally emitted by an object, given in units of temperature
[2] dry-bulb temperature is the temperature of air measured by a thermometer freely exposed to the air but shielded from radiation and moisture
[3] http://jena.sourceforge.net

For example, the rule used to determine whether two *SpatialThings* intersect is:

```
(?x spc:intersects ?y) <-
   (?x rdf:type spc:SpatialThing) (?y rdf:type spc:SpatialThing)
   (?x spc:locatedAt ?xExt) (?y spc:locatedAt ?yExt)
   spatiallyIntersects(?xExt,?yExt).
```

The rules use special builtins that were created for each of the eight relations. The builtins use the JTS topology suite [11] to determine if a specific relation holds between two spatial things. It first converts spatial things into JTS geometry objects and then calls the appropriate method on the geometry objects to perform the check.

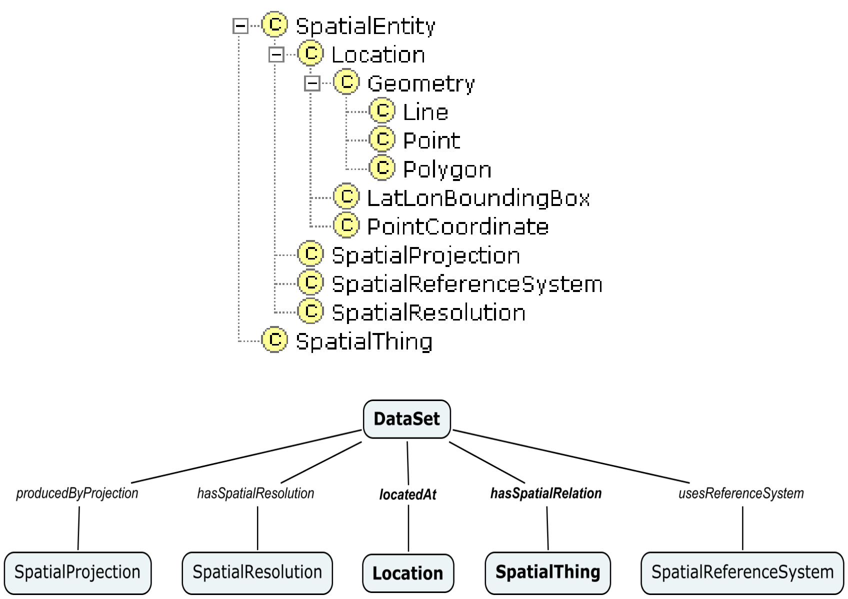


Fig. 4. The spatial ontology and representing spatial properties of observations in SWAP

The *swap-time* ontology incorporates the OWL-Time [16] ontology to represent and reason about the temporal aspects of data (Fig. 5). OWL-Time considers a temporal entity to be either a temporal instant or a temporal interval. As with the spatial reasoner an additional set of temporal rules, based on the COBRA temporal reasoner [7], specify temporal relations.

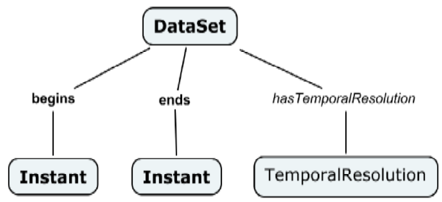


Fig. 5. Representing the temporal properties of a DataSet

For example, the two rules for determining whether a time instant is inside a time interval are:

```
 (?x tme:inside ?y) <-
    (?x rdf:type tme:InstantThing),
    (?y rdf:type tme:IntervalThing),
    (?y tme:begins ?beginsY), (?y tme:ends ?endsY),
    (?beginsY tme:before ?x), (?x tme:before ?endsY).

(?x tme:before ?y) <-
     (?x rdf:type tme:InstantThing),
     (?x tme:inCalendarClockDataType ?timeX),
     (?y rdf:type tme:InstantThing),
     (?y tme:inCalendarClockDataType ?timeY),
     lessThan(?timeX,?timeY).
```

where `tme` is the name space of the OWL-Time ontology. The first rule stipulates that a time instant x is within a time interval y if the starting time of y is before x, and x is before the ending time of y. The second rule uses the `lessThan` builtin to determine whether the time value of a time instant x is before the time value of another time instant y.

4.3 The uncertainty dimension

SWAP takes a Bayesian probability [28] approach to represent and reason about uncertainty on the Sensor Web. Bayesian probability is well suited for dealing with uncertainty on the Sensor Web: where no complete theory is available; where it exists it might be too tedious or complex to incorporate all the required observations; or where all the necessary observation data is not available [28].

The occurrence of natural phenomena is sometimes difficult to detect. However, certain phenomena sometimes exhibit consistent symptoms that are more easily detected and can serve as an indicator for the occurrence of the phenomena. The analysis of observations from multiple sensors may be required to determine the existence of the symptoms of specific phenomena. A Bayesian Network can be used to determine the probability of the occurrence of a phenomenon given one or more observable symptoms.

In such a Bayesian Network two types of discrete random variables are required:

- **Observable event variables:** represents the occurrence of a symptom of a phenomenon and is a qualitative measure for an observation. The variable must specify the entity, the characteristic of the entity being observed, as well as the property that contains the numerical value for the observation. The states are predefined numerical ranges, corresponding to qualitative descriptions. For example, wind speed is often used as an indication of the extent of a storm: from 6 to 49 km/hr is a breeze; 50 to 89 km/hr is a gale; 90 to 117 km/hr is a storm and speeds greater than 118 km/hr is indicative of a hurricane[4]. Observation values can be used to populate observable event variables.

- **Inferred event variables:** represents the occurrence of a phenomenon, e.g. a hurricane. A phenomenon is represented as a subclass of *Phenomenon* in the *swap-theme* ontology. When a phenomenon is detected, an instance of the appropriate class is created. These events are inferred from observable events or other inferred events. Even though these variables are intended for representing the occurrence of a phenomenon, they can be used to represent any event that is not easily or directly measurable.

An occurrence of an observable event is determined by evaluating measurements of some observed property of an observed entity, e.g. the speed of the wind above a certain threshold results in the occurrence of a "strong wind" event. These observable events are used to infer the probability of the occurrence of other events, e.g. a very strong wind is a symptom of a hurricane event. Thus, by analysing one or more measurements certain phenomena can be detected, e.g. a wind speed above 118 km/hr and an air pressure lower than 97.7 kPa can be considered to be symptoms of a hurricane event[5]. A simple Bayesian Network for determining the probability that a hurricane is occurring is shown in Fig. 6. The proposed Bayesian Network model assumes that all variables are discrete and represent events that occur at the same time and space. A limitation of the current model is that it does not cater for the influence of past or future events, or the influence of events occurring at different locations.

An ontology to represent Bayesian Networks.
The *swap-uncertainty* ontology, shown in Fig. 7 extends the BayesOWL [12] ontology. The BayesOWL ontology proposes five classes to represent a Bayesian Network, i.e. *ProbObj*, which could either be a *CondProb* or a *PriorProb*, *Variable* and *State*. A *ProbObj* has a probability value (*hasProbValue*) of some variable

---

[4] Using the Beaufort scale from http://www.hwn.org/home/bws.html
[5] Using the Saffir-Simpson Hurricane Wind Scale, from http://www.nhc.noaa.gov/sshws.shtml

(*hasVariable*) being true. In BayesOWL a *Variable* represents whether an instance is a member (*rdf:type*) of the specified class (*hasClass*) with one of two states, either *True* or *False*.

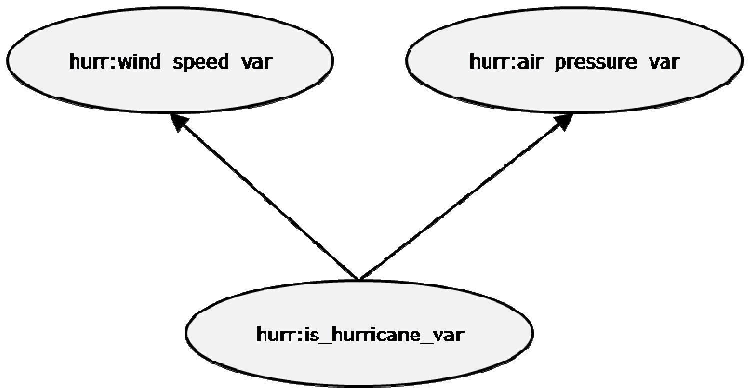


Fig. 6. A Bayesian Network to determine the occurrence of an hurricane from air pressure and wind speed observations

One extension to the BayesOWL ontology is the specialization of the *State* class to allow for user defined *DiscreteStates*. The *DiscreteRangeState* could be a numeric interval for numerical data type properties or a *SingleNumericState* for single numeric values.

The *swap-uncertainty* ontology provides support to represent one or more Bayesian Networks (BN). Each node in the BN represents either an observation or an inferred variable. An observation variable represents the observation value (*hasValueProperty*) for some observed property (*observesProperty*) of the observed entity (*observesEntity*). An inferred variable represents the occurrence of some phenomena (*hasClass*). The *influencedBy* property is used to specify the variables that influence the state of the variable.

SWAP uses the BNJ toolkit for internal representation and inferencing. Bayesian Network tools in Java (BNJ)[6]. BNJ is an open source Java toolkit for developing applications that use Bayesian Networks. It provides a visual Bayesian Network editor and viewer, a graph representation model for representing and manipulating a

[6] http://bnj.sourceforge.net

BN, a number of inference engines, as well as learning algorithms for constructing a Bayesian Network from data.

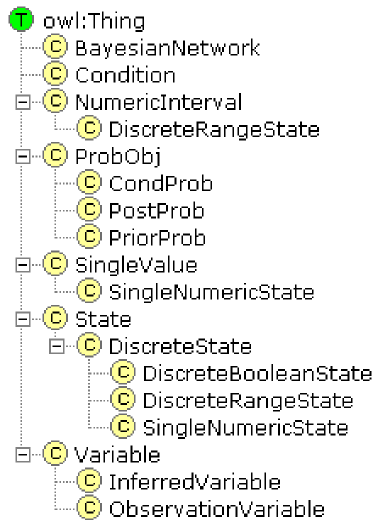


Fig. 7. A fragment of the SWAP uncertainty ontology

A *BayesianNetwork* instance uses the states of observed variables (observation instances) to make inferences about whether a phenomena has occurred (inferred variables). If a phenomena has occurred then an instance of the corresponding phenomena, which contains the corresponding location and time of the observations, is created. A schema ontology containing the BN and observation instances from the knowledge base are provided to the inference engine. The BN is first extracted from the schema ontology and used to create a BNJ graph model. The URIs of the variables and their states are used as the variable and state names in the BNJ graph model to ease the mapping of variables and states between the ontology and the graph model.

In this way the uncertainty reasoner dynamically populates user defined bayesian networks with observable events, performs inferencing on these events and determines and records the occurrence of other events.

4.4 System ontologies

SWAP provides three technical ontologies, i.e. *swap-data*, *swap-agent* and *swap-task* that provide representational support to describe the system entities that are required for hosting and transmitting observations, and for executing algorithms and theories.

The *swap-data* ontology provides descriptions of different data structures that can be exchanged between agents. This includes coverage (image) and feature data as well as units of measure.

Representing agents

The *swap-agent* ontology provides support for representing an agent, the service it hosts and the interaction protocol required to invoke the service. It provides support for representing the six different types of agents specified in the SWAP abstract architecture (Fig. 8) [21]. These are data provider (Sensor) agents, processing or data transformation (Tool) agents, modeling (Modeling) agents and coordination (Workflow) and application (Application) agents.

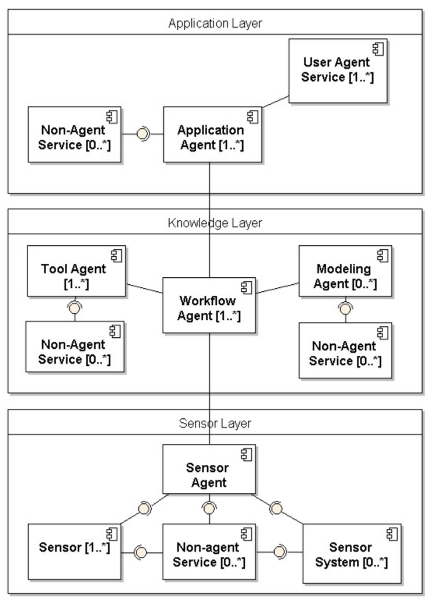


Fig. 8. The SWAP abstract agent architecture [21]

Each agent type has a corresponding service description with a set of common attributes that capture the conceptual functionality of the service. Sensor Agents provide a description of the observations that they provide, while Tool and Modeling Agents provide a description of the data processing algorithms and prediction models that they respectively provide. Service description attributes are grouped into the four different conceptual systems, i.e. spatial, temporal, thematic and uncertainty, and are specified using concepts from the appropriate top level ontology. Service descriptions also contain service invocation information in the form of input and output mappings.

A request and a response message template is used for invoking and interpreting the response of the service. The request message template specifies all service invocation parameters, which may be mandatory or optional parameters that have default values. Users populate mandatory parameters and may also specify optional parameters for finer control of the service. These message templates are used to dynamically invoke a service and to consume and interpret its results. This bridges the gap between service selection and use, i.e. once a suitable service has been identified it can be dynamically invoked and its results can be dynamically interpreted.

Representing services and workflows

The *swap-task* ontology is based on OWL-S [32], and provides algorithmic primitives to assemble multiple agents into executable agent workflows. An agent is represented as atomic processes and OWL-S algorithmic constructs are used to assemble multiple agents into appropriate sequences of invocations or composite processes. The main extension to OWL-S is a process to agent mapping that allows OWL-S processing steps to be transformed into agent invocations at runtime. The mapping specifies request and response templates that are used to transform each processing step into an appropriate request and response message used to invoke an agent and to interpret its response.

The technical ontologies provide support for describing the services offered by different agents and the agent interactions used to invoke these services. Support is also provided for constructing complex information processing chains or workflows that may be stored, shared and executed on demand. Since service descriptions and data models are captured within shared ontologies, they become dynamic entities that can be accessed, queried and modified at runtime. Selected services can be assembled into different configurations to form complex executable workflows that may be deployed as new composite services. This approach facilitates interoperability between agents, and between agents and humans. It also allows for data models and service offerings to change, and evolve naturally with minimal impact and without having to re-engineer the system.

Together, the technical and conceptual ontologies allow SWAP users to represent complex information processing chains or workflows. Users search semantic agent service descriptions and identify appropriate sensor data sets, algorithms and models to apply to these data sets. Once the appropriate agents are identified, users use the algorithmic constructs in the *swap-task* ontology to specify a processing workflow that assembles different agent services in an appropriate sequence for execution. Each workflow represents new functionality in the system. A workflow can also be deployed on a Workflow Agent where it can be executed on demand. Since a workflow is fully specified and executed from its OWL-S specification, the appropriate ontologies (which contain the workflow) can be shared, downloaded and executed locally. Furthermore, once the workflow is downloaded it can be easily modified and executed locally by SWAP users. A workflow is represented as a composite process, which means that it can be incorporated into other composite processes (workflows). This allows for reuse of existing workflows within other workflows and for creating and managing large and complex nested workflows. Currently, workflows are created and modified manually via an ontology editor. However, given that the semantics of both the conceptual and the technical aspects of each service are specified in the service description, this provides a sound foundation for automating workflow composition.

5 Conclusion

We have introduced the notion and proposed knowledge representation requirements and potential benefits of a Knowledge Driven Sensor Web. We contend that a semantic infrastructure and formal software modeling and engineering abstractions are both equally important to manage data and knowledge dynamisms as well as system and application dynamism. We propose an ontology driven multi-agent system approach to constructing such a system. A key limitation in agent based approaches is the lack of a comprehensive semantic infrastructure that includes support for representing uncertainty, theories and beliefs and support for representing agents, services and tasks. A semantic infrastructure that deals with these limitations was described. A novel aspect is the introduction of the additional modeling dimension of uncertainty which can be used for representing and applying subjective theories. The *swap-uncertainty* ontology incorporates Bayesian probability, which is widely used in practical applications to represent degrees of belief, and allows for the incorporation of Bayesian Networks to represent different theories of cause and effect relations between events in the physical world. The nature and availability of sensor data, the accuracy and completeness of the theory that underpins the choice, and the sequence of the processing steps may contribute an additional element of uncertainty. The information produced by workflows is frequently approximations or best guesses. The incorporation of uncertainty allows end users to better understand the quality of information generated within the Sensor Web.

There are many avenues for future work. The relation of this work to the trend in the Semantic Sensor Web community towards linked data [9][4][18] warrants further investigation. Another avenue is the extension of the uncertainty model to capture and reason about relations between past, current and future events and events occurring at different locations.

REFERENCES

1. Agarwal, P. (2005), 'Ontological considerations in GIScience', International Journal of Geographical Information Science 19(5), 501-535.
2. Athanasiadis, I.N.; Milis, M.; Mitkas, P.A. & Michaelides, S.C. (2005), Abacus: A multi-agent system for meteorological radar data management and decision support, in 'ISESS-05: International Symposium on Environmental Software Systems, Sesimbra, Portugal, May 2005'.
3. Balazinska, M.; Deshpande, A.; Franklin, M.J.; Gibbons, P.B.; Gray, J.; Hansen, M.; Liebhold, M.; Nath, S.; Szalay, A. & Tao, V. (2007), 'Data Management in the Worldwide Sensor Web', IEEE Pervasive Computing 6(2), 30-40.
4. Barnaghi, P. & Presser, M. (2010), Publishing Linked Sensor Data, in ‘Kerry Taylor, Arun Ayyagari, David De Roure (Eds.): The 3rd International workshop on Semantic Sensor Networks 2010 (SSN10) in conjunction with the 9th International Semantic Web Conference (ISWC 2010), 7-11 November 2010, Shanghai, China'.
5. Biswas, P.K.; Schmiedekamp, M. & Phoha, S. (2006), 'An agent-oriented information processing architecture for sensor network applications', Int. J. Ad Hoc Ubiquitous Comput. 1(3), 110-125.
6. Bröring, A.; Echterhoff, J.; Jirka, S.; Simonis, I.; Everding, T.; Stasch, C.; Liang, S. & Lemmens, R. (2011), 'New Generation Sensor Web Enablement', Sensors 11(3), 2652-2699.
7. Chen, H.L. (2004),'An Intelligent Broker Architecture For Pervasive Context-Aware Systems', PhD thesis, University of Maryland, Baltimore County.
8. Compton, M.; Henson, C.; Neuhaus, H.; Lefort, L. & Sheth, A. (2009),A Survey of the Semantic Specification of Sensors, in 'Kerry Taylor, Arun Ayyagari, David De Roure (Eds.): The Proceedings of the 2nd International Workshop on Semantic Sensor Networks (SSN09), in conjunction with the 8th International Semantic Web Conference (ISWC 2009), Washington DC, USA, October 26, 2009'.
9. Corcho, O. & Garcia-Castro, R. (2010), 'Five challenges for the Semantic Sensor Web', Semantic Web 1(1-2), 121-125.
10. Cox, S. (2006),'Observations and Measurements. Discussion Paper. OGC 05-087r3 Version 0.13.0 Observations and Measurements'.
11. Davis, M. (2007), Secrets of the JTS Topology Suite, ‘Free and Open Source Software for Geospatial 2007 (FOSS4G2007), Victoria, Canada, 24-27 September 2007'.
12. Ding, Z.; Peng, Y. & Pan, R. (2006), Soft Computing in Ontologies and Semantic Web, Springer-Verlag, chapter BayesOWL: Uncertainty Modeling in Semantic Web Ontologies, pp. 3-29.
13. Gaspari, M. (1998), 'Concurrency and knowledge-level communication in agent languages', Artif. Intell. 105(1-2), 1-45.
14. Gibbons, P.B.; Karp, B.; Ke, Y.; Nath, S. & Seshan, S. (2003), 'IrisNet: An Architecture for a Worldwide Sensor Web', IEEE Pervasive Computing 2(4), 22-33.
15. Hakimpour, F.; Aleman-Meza, B.; Perry, M. & Sheth, A. (2006), Data Processing in Space, Time and Semantics Dimensions, in ‘Terra Cognita Workshop - Directions to the Geospatial Semantic Web, A workshop of the 5th International Semantic Web Conference ISWC 2006, November 5-9, 2006, Athens, Georgia, USA'.
16. Hobbs, J.R. & Pan, F. (2004), 'An ontology of time for the semantic web', ACM Transactions on Asian Language Information Processing (TALIP) 3(1), 66-85.

17. Jennings, N.R. (2001), 'An agent-based approach for building complex software systems', Commun. ACM 44(4), 35-41.
18. Kessler, C. & Janowicz, K. (2010), Linking Sensor Data - Why, to What, and How?, in ‘Kerry Taylor, Arun Ayyagari, David De Roure (Eds.): The 3rd International workshop on Semantic Sensor Networks 2010 (SSN10) in conjunction with the 9th International Semantic Web Conference (ISWC 2010), 7-11 November 2010, Shanghai,China'.
19. Mennis, J.; Peuquet, D. & Qian, L. (2000), 'A conceptual framework for incorporating cognitive principles into geographical database representation', International Journal of Geographical Information Science 1414(6), 501-520.
20. Moodley, D. (2009),'Ontology Driven Multi-Agent Systems: An architecture for Sensor Web applications', PhD thesis, School of Computer Science, University of KwaZulu-Natal, Durban, South Africa.
21. Moodley, D., and Simonis, I. (2006), A new architecture for the sensor web: the SWAP-framework, in ‘Semantic Sensor Networks Workshop, A workshop of the 5th International Semantic Web Conference ISWC 2006, November 5-9, 2006, Athens, Georgia, USA.
22. Moodley, D.; Vahed, A.; Simonis, I.; McFerren, G. & Zyl, T.V. (2008), 'Enabling a new era of Earth observation research: scientific workflows for the Sensor Web', Ecological Circuits 1, 20-23.
23. Nolan, J. (2003),'An agent-based architecture for distributed imagery and geospatial computing', PhD thesis, George Mason University.
24. Open GIS Consortium, I. (May 5, 1999),'OpenGIS Simple Features Specification For SQL Revision 1.1', Technical report, OpenGIS Project Document 99-049.
25. Patel-Schneider, P.F. & Horrocks, I. (2007),'OWL 1.1: Web Ontology Language Overview'.
26. Perry, M.; Hakimpour, F. & Sheth, A. (2006),Analyzing theme, space, and time: an ontology-based approach, in 'GIS '06: Proc. 14th annual ACM international symposium on Advances in geographic information systems', ACM Press, New York, NY, USA, pp. 147-154.
27. Raskin, R.G. & Pan, M.J. (2005), 'Knowledge representation in the semantic web for Earth and environmental terminology (SWEET)', Computers & Geosciences 31(9), 1119-1125.
28. Russell, S. & Norvig, P. (2003), Artificial Intelligence: A Modern Approach, Prentice-Hall, Englewood Cliffs, NJ.
29. Russomanno, D.J.; Kothari, C.R. & Thomas, O.A. (2005), Building a Sensor Ontology: A Practical Approach Leveraging ISO and OGC Models, in 'IC-AI', pp. 637-643.
30. Shadbolt, N.; Berners-Lee, T. & Hall, W. (2006), 'The Semantic Web Revisited', IEEE Intelligent Systems 21(3), 96-101.
31. Terhorst, A.; Moodley, D.; Simonis, I.; Frost, P.; McFerren, G.; Roos, S. & van den Bergh, F. (2008), Geosensor Networks, Lecture Notes in Computer Science, Volume 4540/2008, Springer-Verlag, chapter Using the Sensor Web to Detect and Monitor the Spread of Vegetation Fires in Southern Africa, pp. 239-251.
32. The OWL-S Coalition (2004),'OWL-S: Semantic Markup for Web Services'. Available online at: http://www.w3.org/Submission/OWL-S/
33. van Zyl, T.L.; Simonis, I. & McFerren, G. (2009), ''The Sensor Web: systems of sensor systems', International Journal of Digital Earth 2:1, 16-30.
34. Wijngaards, N.J.E.; Overeinder, B.J.; van Steen, M. & Brazier, F.M.T. (2002), 'Supporting internet-scale multi-agent systems', Data Knowl. Eng. 41(2-3), 229-245.
35. Zambonelli, F.; Jenning, N.R.; Omicini, A. & Wooldridge, M.J. (2001), Agent-oriented software engineering for Internet agents, Springer-Verlag, London, UK, pp. 326-346.

Aggregating Linked Sensor Data

Christoph Stasch[1], Sven Schade[2], Alejandro Llaves[1], Krzysztof Janowicz[3], and Arne Bröring[145]

[1] Institute for Geoinformatics, University of Münster, Germany
[2] Institute for Environment and Sustainability, Joint Research Centre, Ispra, Italy
[3] University of California, Santa Barbara, USA
[4] ITC Faculty, University of Twente, Enschede, Netherlands
[5] 52°North Initiative for Geospatial Open Source Software GmbH, Germany

Abstract. Sensor observations are usually offered in relation to a specific purpose, e.g., for reporting fine dust emissions, following strict procedures, and spatio-temporal scales. Consequently, the huge amount of data gathered by today's public and private sensor networks is most often not reused outside of its initial creation context. Fostering the reusability of observations and derived applications calls for (i) spatial, temporal, and thematic aggregation of measured values, and (ii) easy integration mechanisms with external data sources. In this paper, we investigate how work on sensor observation aggregation can be incorporated into a Linked Data framework focusing on external linkage as well as provenance information. We show that Linked Data adds new aspects to the aggregation problem, e.g., whether external links from one of the original observations can be preserved for the aggregate. The Stimulus-Sensor-Observation (SSO) ontology design pattern is extended by classes and relations necessary to model the aggregation of sensor observations.

Keywords: Sensor Aggregation, Semantic Enablement, Linked Data

1 Introduction

Sensor observations are collected with a specific purpose in mind and, therefore, measuring follows strict procedures and spatio-temporal scales [1]. While the same device, e.g., a thermometer, can be used to measure soil and air temperature, both follow different procedures and their results cannot be combined. Similar issues hold for fine dust $(PM10)^6$ measurements, where data coming from rural monitoring stations has to be distinguished from data produced by sensors located in urban areas, particularly at major roads [2]. Consequently, the rich observation data gathered by today's public and private sensor networks is difficult to reuse aside of the initially intended context. We hope to boost the use of observation results and the number of innovative observation-based applications by providing mechanisms for (i) spatial, temporal, and thematic aggregation of measured values, and (ii) easy integration mechanisms with other data sources.

[6] The notation PM10 is used to describe fine dust particles of 10 micrometers or less.

Building up on our previous work on exposing standardized observation data as Linked Data [3], this paper introduces the next steps towards opening up sensor observations to new usage scenarios: the aggregation of observations and exposing them as Linked Sensor Data. Having temporal aggregates (e.g., yearly averaged fine dust measures), spatial aggregates (e.g., fine dust concentration in the Münsterland region in Germany), thematically aggregated observations (e.g., blizzards, landslides, or forest fires), and their combinations available, makes linking more attractive and opens environmental information to new user communities. On the one hand, observations may be connected to particular features of interest in the Linked Data cloud. On the other hand, hubs such as DBpedia may directly refer to aggregated observations, e.g., an entry about the German city of Münster and its surrounding areas by referring to recent and average weather conditions, or air quality measures.

The main contributions of this paper are threefold. We (i) present a Linked Data model for aggregated sensor data, (ii) discuss the effects of aggregation on links from and to observations, and (iii) outline the role of provenance in this setting. The implementation of the extensions discussed in this paper are ongoing and the 52°North semantics community[7] plans to release an updated prototype in fall 2011.

The remainder of this paper is structured as follows. In section 2.1, we introduce the concept of aggregated observations and provide background information about Linked Sensor Data and provenance information in observations. Section 3 discusses the implication of aggregation on Linked Sensor Data. Here, we present the required extensions to our Linked Data model for observations. Additional investigations address the effects of aggregation on external linking, and issues on data provenance. In section 4, we set our work in relation to current efforts to provide observations as Linked Data and to provide provenance information in observation data. The paper concludes with a summary and an outline of the remaining steps for implementing aggregated observations as Linked Data; see section 5.

2 Background

In this section, we provide a brief overview on related work. At first, we introduce definitions and related work about aggregation of observations. Second, we introduce the concepts of Linked Data. Finally, we describe related work about provenance of sensor data.

2.1 Aggregation of Observations

Aggregation of observations in space and time is essential to derive information that is useful for a certain application purpose and to integrate observation data with differing spatio-temporal resolutions. Yet, spatio-temporal aggregation of

[7] Implementations and documentations can be found at `http://52north.org`.

observations in the Linked Data context has not yet been addressed. However, in other communities, e.g., the database community or in environmental sciences, spatio-temporal aggregation has been a research topic for years and is sometimes also referred to as scaling of observations and environmental models. Vega and Lopez [4] give a comprehensive survey on spatio-temporal aggregation methods for databases. Besides simple aggregation, complex statistical models might also be applied as described for the domain of soil sciences by Bierkens et al. [5]. Spatio-temporal aggregation processes for observation data are not yet available on the Web and, therefore, recent approaches demonstrate how to tackle this challenge. A spatio-temporal aggregation service that can be used to provide such aggregation functionality on the Sensor Web has been introduced in our previous work [6].

In this paper, we largely follow the definition of aggregation[8] by Jeong et. al. [7]. During an *aggregation process*, the observations are grouped by a *grouping predicate*, e.g., by a spatial predicate which is defined by the polygon representing the area of a city, or by a temporal predicate defined by the time period of a month. After grouping, an *aggregation function* is applied that computes a single value, an *aggregate*, for the result values of an observation group. The aggregation function might be linear (e.g., MEAN), but also non-linear (e.g., MEDIAN, or areal fraction of spatial blocks where the concentration of a pollutant exceeds a critical level) [8]. The grouping predicate does not necessarily have to be the target spatial or temporal extent of an aggregated observation. Considering the example of a block kriging method [9], for every aggregate of a spatial block, all measurements are taken into account and not just the ones laying in the extent of the block. Similarly, for temporal aggregation, moving windows might be used to aggregate values to time periods that also include the values before and after a certain period.

Besides spatio-temporal aggregation as introduced above, extracting high level events from observations is also done by aggregating observations. Treating the high level events as observations again enables an easy integration into existing infrastructures and tools. Considering the blizzard example as described in [10], the event of a blizzard can also be modeled as an observation. The blizzard is an aggregate of several observations indicating heavy snowfall, very low temperatures, and high wind speed. This example demonstrates that the grouping predicate is not merely spatial or temporal, but also contains predicates on the result values of the observations (e.g., heavy snowfall). We thus refer to this kind of aggregation as *thematic aggregation* of observations. However, the observations are still aggregated spatially or temporally as the blizzard is observed at a region in space and for a period in time.

[8] Aggregation might be also referred to as complex entity with parts. In case of observations, this might be a collection of observations where the non-aggregated observations are parts of the aggregated observation collection. However, in our work we consider aggregation as described in this paragraph and commonly used in environmental sciences.

2.2 Linked Data

For aggregation of observations a mechanism that helps to retrace the original observations and sensors from the aggregated observations is important. Linked Data [11] provides a promising paradigm to provide such a mechanism, as the original observations and the aggregates can be easily linked with clear semantics. Linked Data proposes unique identifiers for data in the Web, links between them, and relies on the Resource Description Framework (RDF) [12]. The most common query language for RDF is SPARQL [13]. SPARQL has similar capabilities as query languages for relational databases, but works by matching graph patterns and is optimized for RDF triple stores, such as Sesame or Virtuoso. Within the last years, Linked Data has become the most promising vision for the Future Internet and has been widely adopted by academia and industry.

Several approaches for Linked Sensor Data in the Web are already available [14,15,16]. They describe, how to identify sensor resources using URIs, how to link them with clear semantics and how to expose the sensor data in the Web. However, the issue of spatio-temporal aggregation, e.g. how aggregation affects the links from and to observations, is not yet addressed. In our previous work [3], we developed a standards-based approach to expose sensor metadata and observations stored in a Sensor Observation Service (SOS) [17] to the Semantic Web by following Linked Data principles and providing dereference-able HTTP URIs for sensors, observed properties, features of interest, and observations, link them (to external sources), and expose their semantics using the SSO ontology [18]. In this work, we extend our previous work on Linked Sensor Data to support aggregated observations.

2.3 Provenance in Observation Data

There are several approaches available for providing provenance information in the Web. The W3C's Provenance Incubator Group[9], predecessor of the new Provenance Working Group[10], compiled a list of requirements to support provenance in RDF, which includes for example that every observation should have an URI identifier [19]. Based on these requirements, the Provenance Vocabulary has been defined[11] that can be used in the Web to provide provenance information for Linked Data [20]. Similar to the Provenance Vocabulary, the Open Provenance Model[12] (OPM) defines nodes and edges to create provenance graphs that allow to retrace the creation of an item back to its origin. The nodes can be *artifacts*, *processes* and *agents* whereas the edges between nodes can be defined as the causal relationships *used*, *wasGeneratedBy*, *wasControlledBy*, *wasTriggeredBy*, and *wasDerivedFrom*. The graphs can be serialized in different data formats like XML.

[9] `http://www.w3.org/2005/Incubator/prov/charter`
[10] `http://www.w3.org/2011/01/prov-wg-charter`
[11] `http://sourceforge.net/apps/mediawiki/trdf/index.php?title=Provenance_Vocabulary`
[12] `http://openprovenance.org/`

Besides general approaches for provenance information in the Web, providing provenance information in Linked Sensor Data has recently gained attention. Provenance of sensor data can be defined as information about the source of the sensor data as well as information about transformations applied to the original data [21]. Patni et al. [10] propose an approach for provenance in Linked Sensor Data and define the capabilities of the sensor, the spatio-temporal parameters of the observation, and the measurement value as relevant sensor provenance information. Liu et al. [22] introduce a provenance aware virtual sensor system based upon the OPM. Using the OPM for their virtual sensors enables the description of (i) fetching processes for sensor data streams; (ii) workflow execution like data transformation of raw measurements; and (iii) user interaction with a web application that allows to manage the virtual sensors. In another approach, Park and Heidemann [21] defined their own provenance model that (for sensor data) is more comprehensive than the OPM. Among other things, this alternative model allows the definition of access control for sources. Similar to the approach of Liu and colleagues, the sensor data is annotated with additional provenance metadata. Our approach will show how most of relevant provenance information is already provided in our Linked Sensor Data and how the links can be mapped to provenance relationships as defined in the OPM.

3 Aggregation of Observations in the Linked Data Cloud

In this section we introduce an approach to enable the aggregation of observations in the Linked Data cloud. First, we present an extension of the Stimulus-Sensor-Observation (SSO) ontology design pattern [18]. Next, we illustrate how the change of observation properties during aggregation affects the links from and to observations in the cloud. Finally, we describe how provenance information pointing back to the original observations can be provided.

3.1 Extension of the SSO Design Pattern

Following our previous work [3], we use an intermediate Linked Data model for exposing sensor observations. It was derived from an ontology developed by the W3C SSN-XG [23], namely the Stimulus-Sensor-Observation (SSO) ontology design pattern [18]. The SSO pattern forms a generic and adaptable starting point for the development of sensor ontologies as well as Linked Data vocabularies.

Figure 1 shows the classes and relations from the pattern extended by the Linked Data model for sensor data, and the new elements that have been added in order to account for aggregation. In a nutshell, we reuse the following definitions:

- *FeatureOfInterest*: entity that comprises observable properties.
- *ObservedProperty*: property that inheres in a feature of interest.
- *ObservationCollection*: set of observations, grouped by a distinct criteria.
- *Observation*: (social) construct that connects observed properties with sensors, sensing results, and sampling times.

– *SamplingTime*: time instant or interval at which an observation was made.
– *Result*: symbol representing an observed value.
– *Sensor*: entity that performs observations.
– *Procedure*: description that specifies how observations have to be carried out

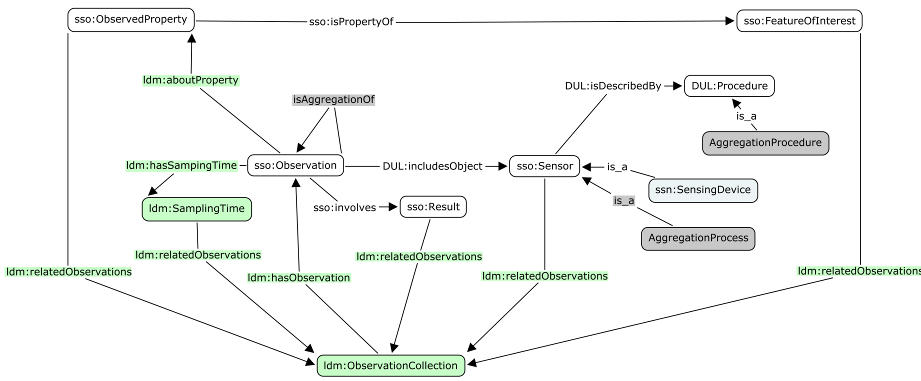


Fig. 1. Partial concept map with the classes and relations of the Linked Data model for aggregated observations, extensions highlighted in grey. The prefix *sso* indicates elements taken from the SSO pattern, DUL indicates elements of DOLCE Ultra Light, *ssn* those of the W3C Semantic Sensor Network ontology (in pale blue), and *ldm* elements of the Linked Sensor Data model.

In order to account for aggregated observations as Linked Data, we extend the SSO pattern with the following elements:

– *isAggregateOf*: a relation that allows one observation to be aggregated out of others; e.g., an observation of daily $PM10$ concentration being an aggregate over hourly measures, or an observation of $PM10$ in Münster, Germany being an aggregate over various Point of Interest (POI) measures.
– *SensingDevice*[13]: a sensor, which is a physical measuring device; e.g., a particular air sampler including a special filter $PM10$.
– *AggregationProcess*: a sensor, which implements a concrete aggregation procedure (see below), for example the process that calculates regional $PM10$ concentrations based on several $PM10$ concentration observations and additional calibration parameters.

[13] The concept of a *SensingDevice* is also captured as part of the W3C SSN-XG ontology. However, it is not part of the SSO pattern, which is applied in our work. We decided to introduce the *SensingDevice* in particular opposed to the notion of the *AggregationProcess* in order to stress the difference between a single physical measurement instrument and the aggregation process that combines multiple sensory inputs to a new observation.

– *AggregationProcedure*: the specific procedure used for aggregating several observations into one; e.g., calculating the MEAN of 24 hourly observations of $PM10$ concentration, or a Kriging interpolation method

The relations between the classes presented in Figure 1 act as links in our model and define the multiple navigation paths and external references; see also [3]. The above mentioned extensions allow for the generation of aggregated observations together with an explicit mentioning of the applied aggregation method, such as MIN, MEAN, or MAX calculations over a temporal series. This also allows for linking aggregated observations back to finer grained observations (discussed in Section 3.3). This new model can be used as URI scheme and query filter to enable the Restful Linked Data SOS to serve aggregated data as well.

3.2 Effects on Links from and to Observations

Aggregating linked observations affects the links from and to the observations. Questions like 'Are the links to a feature of interest still valid, if observations taken at specific points are spatially aggregated to an area?' or 'Which new links can be established after aggregation of an observation?' need to be answered. First of all, the links which are defined in our observation ontology need to be checked for consistency and changed, if necessary[14]. Table 1 shows examples of objects (i.e., link targets) of the links from observations before and after aggregation of point observations to an area in space and a period in time.

Independent of the concrete example, for each aggregation the target of the *ldm:hasSamplingTime* link changes from an instant in time (original observations) to a period in time (aggregated observations), if the observations are aggregated temporally. Also, the *DUL:includesObject* link will always point from the aggregated observation to an instance of an *AggregationProcess* instead of pointing to a specific *SensingDevice* from the original observations. In environmental applications, the *ObservedProperty* is usually a continuous phenomenon, which is sampled at certain locations in space or time, e.g., $PM10$ concentration. If only spatial and/or temporal aggregations are applied, the *ldm:aboutProperty* remains the same. In case of a thematic aggregation (see Section 2.1), the *ObservedProperty* changes. An example is the blizzard as a combination of high wind-speed, heavy snowfall, and low temperatures: the *ObservedProperty* of the blizzard observation is the phenomenon of the blizzard, whereas the original observations point to the phenomena wind-speed, snowfall and surface temperature. Similar examples could be constructed for landslides or forest fires.

Though the *ObservedProperty* might be unchanged during an aggregation process, the *sso:isPropertyOf* link changes, if the observations are aggregated in

[14] Here, changing links means that triples of the original observations might be removed and replaced by other triples in the aggregated observations for the same relationship. For example, the hasSamplingTime relationship usually links to a point in time in the original observations, but to a time period in the aggregated observations, if the observations are aggregated temporally.

Link in Ontology	**Object Before Aggregation**	**Object After Aggregation**
ldm:hasSamplingTime	TimeInstant *(08/05/2011; 23:15 CEST)*	TimePeriod *(one day)*
DUL:includesObject	SensingDevice *(air sampler)*	AggregationProcess *(block kriging of PM10 measures)*
ldm:aboutProperty	ObservedProperty *(PM10)*	ObservedProperty *(PM10)*
sso:isPropertyOf	SamplingPoint *(N 51 57.466 E 007.37.433)*	GeospatialRegion *(area of Münster)*
sso:involves	MeasurementValue *(110)*	Aggregate derived from multiple Measurement Values *(70)*

Table 1. Object of links from observations before and after an aggregation of point measurements to an area in space and period in time; examples are given in parenthesis.

space. For example, aggregating the point measurements to an area causes the *FeatureOfInterest* to change from a sampling point to an upper level feature like the area of the city of Münster. Finally, the *sso:involves* link points to an aggregate computed during the aggregation process. Originally, the *sso:involves* link has pointed to the measurement values from the source observations. Besides changing the original links, additional links might be added pointing to the aggregated observation. As introduced in our model, the *isAggregateOf* link points from an aggregated observation to the original observations. Furthermore, other observation collections might contain the aggregated observation resulting in new *ldm:hasObservation* links to the aggregated observation. Also, other higher level features like cities, administrative areas, etc. might be linked to the aggregated observations.

Formalizing the changes of links during aggregation is challenging and often domain specific. However, we consider the identification and formalization of such changes as crucial to provide a (semi-)automated aggregation of observations in the Linked Data cloud in future and are currently working on such a formalization.

3.3 Provenance in Aggregated Linked Observation Data

In a Linked Data context where different communities might have interest in interlinking their datasets, it is important to publish trust-able datasets. Provenance information favors trustworthiness of data because users are able to analyze the historic changes and reproduce them [24]. Especially when *aggregating* observations in Linked Data, it is important to be able to retrieve information about the original observations as well as the aggregation process that has been applied. Figure 2 shows a provenance graph that illustrates how provenance information about the aggregation process and original observations is provided in our Linked Data model and how this can be mapped to the concepts of the OPM and the Provenance Vocabulary. The reason to extend our model instead of re-using an existing solution lies in the fact that most of the provenance information needed for sensors and observations is already available, thus we avoid redundancy.

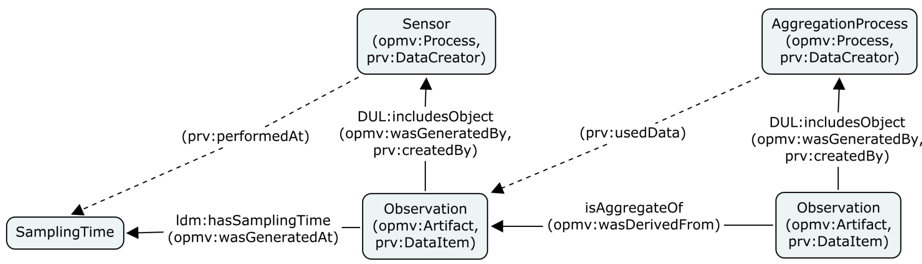


Fig. 2. Provenance graph illustrating how provenance information about the aggregation sensor and original observations is provided in our Linked Data model. The corresponding entries of the OPM vocabulary (with prefix *opmv*) and of the Provenance Vocabulary (with prefix *prv*) are shown in brackets. In case the concepts can not directly mapped but the provenance information can be retrieved following other links in the model, we use dotted lines.

First of all, the *isAggregateOf* allows to trace the aggregated observations back to the original observations. Hence it can be mapped to the *opmv:wasDerivedFrom* relationship in the OPM. Though the *isAggregateOf* relationship cannot be directly mapped to a relationship in the Provenance Vocabulary, it provides basically the information that is provided by the *prv:usedData* link from a *prv:DataCreator* to a *prv:DataItem*. Information about the aggregation process that has created an aggregated observation is provided by the *DUL:includesObject* link from the aggregated observation to the *Aggregation-Process*. This link can be mapped to the *opmv:wasGeneratedBy* relationship in the OPM and to the *prv:createdBy* relationship of the Provenance Vocabulary. The *ldm:hasSamplingTime* attribute provides a link to the time at which the value represents a physical phenomenon in the world. In case of observations taken by a physical sensor this corresponds to the time when the observation has been taken. However, if the observations gathered by physical sensors are aggregated by an *AggregationProcess*, the *SamplingTime* is a time period representing the value for which the aggregate is valid. This is no longer the time when the observation has been produced (time of aggregation). Thus, for aggregated observations, an additional time link might be added providing this information. Similarly, additional links might be provided for the *opmv:wasControlledBy* and the *opmv:used* relationships of the OPM, which we did not yet include, as we focus on retracing the observations and not on the users which are aggregating or using the observations.

4 Discussion

The presented research is in line with the theoretical challenges in Sensor Web research, which have been identified during an expert meeting in 2010 [25], addressing the challenges of interoperability and integration of sensor based system and model based systems. Our extension of the SSO design pattern as described in Section 3.1 allows to expose aggregated observations as Linked Sensor Data. This goes beyond the approaches available for providing Linked Sensor Data [14,15,16] which are focusing on providing non-aggregated observations. In our approach, we follow an observation-centric viewpoint that an aggregated observations is still an observation about a quality in the world and thus can be modeled as such. However, further discussion is needed whether the aggregation process can still be modeled as a sensor or has to be distinguished from the concept of sensors.

Our model also allows to retrace the aggregated observations back to the original observations and to retrieve information about the aggregation process applied, thus providing provenance information about the aggregated observations (see Section 3.3). Instead of providing additional metadata as in other approaches described in Section 2.3, we show how the provenance information can be directly retrieved by using the links established in our Linked Data model for (aggregated) observations. For example, Patni et. al. [10] present an approach for provenance in Linked Sensor Data where a separate provenance ontology has been defined. In contrast, we aim to avoid duplication of, for example, information about which sensor has created an observation at what time. This information is already contained in the existing sensor and observation ontologies. We rather show how the relationships of the observation ontologies providing this information can be mapped to relationships of well-established provenance models like the OPM or the Provenance Vocabulary. In order to enable the integration of observations in tools relying on this common provenance models, the observations can either easily be translated to such models or additional triples can be added in the observation set. However, in both approaches, this causes redundant information which might cause problems dealing with large datasets which is common in environmental sciences. Opposed to the general approach for providing provenance information, e.g., about triples in the Linked Data cloud [20], we do not yet consider provenance information about the instances of objects and links according to our Linked Data model, e.g. *Who has created an observation triple in the Linked Sensor Data at which time*. To provide such information, we think that the general approaches for data provenance in the Web can be utilized.

Both, sensor observations and aggregates provide estimations for physical phenomena occurring in the world. As it is not possible to observe all relevant aspects in reality, observations can only represent reality to a certain degree and thus are uncertain about reality. In studies dealing and using observations, it is crucial to account for the uncertainty. This is usually referred to as uncertainty propagation [26]. Aggregation is one mean to adjust the uncertainty in estimations. The more the data is aggregated, the less uncertainty is in the data. At the moment, we do not yet explicitly account for uncertainty in the presented work. Investigations how uncertainty can be propagated in observation processing workflows in the Web are currently ongoing within the European research project UncertWeb[15] [27]. We plan to adopt their approaches and add the uncertainty to our Linked Data model.

While we are providing the model for exposing aggregated observations as Linked Data and we discuss the effects on links from and to observations during aggregation (Section 3.2), we have not yet addressed the technological aspect of executing aggregation processes on Linked Sensor Data. However, we are currently working on extending our Spatio-Temporal Aggregation Service to also deal with Linked Data serialized as RDF. This also leads to the question to what degree observations should be aggregated before exposing them as Linked Data in order to reduce the amount of triples or whether observations can/should be provided at different aggregation levels as Linked Sensor Data. For example, providing high resolution sensor data as Linked Data might lead to a huge amount of triples which might cause performance problems. Thus, it might be better to aggregate the observations before and then expose them as Linked Data.

5 Conclusions and Outlook

In this paper, we identify the need for spatial, temporal, and thematic aggregations of sensor observations and their propagation as Linked Data for an easy integration with other data sources. Aggregates of sensor observations (e.g., the monthly average fine dust concentration in a city) can be much easier utilized in applications. Facilitating the integration of such aggregated observations by providing them as Linked Data enables their utilization among different applications. We achieve this by: (1) extending the SSO ontology design pattern to accommodate aggregation information and including concepts such as *AggregationProcedure* or *AggregationSensor*; (2) describing how links from point observations change after aggregation (e.g., feature of interest may change from a sampling point to a city area); (3) supporting the provenance information in the model through enabling retraceability to original observations and introduce relations such as *isAggregationOf*.

Our future work will follow these lines. Aside from our ongoing implementation work, we plan to exploit the combination of the proposed approach with event detection mechanisms and stream processing. Therefore, we are planning to combine the extension of the SSO ontology pattern presented in this paper with our previous work on *sensor plug & play* [28]. In that work, we designed a framework that enables the on-the-fly integration of sensors and Sensor Web services by determining the semantic matching between sensor characteristics and service requirements. This framework can also be put to use in on-stream processing for the dynamic fusion of incoming data streams of multiple sensors to produce aggregated observations. This is similar to approaches such as [29],

[15] http://www.uncertweb.org

but also allows the creation of new, combined phenomena. A basic example is the combination of *temperature* and *conductivity* data streams measured by underwater sensors to derive a stream of *salinity* observations.

Furthermore, we are planning to extend our approach developed with representations for uncertainty as described in the Uncertainty Markup Language [30]. Our provenance information currently provides information about the aggregation procedure applied, its implementation, and about the original observations that have been used to derive the aggregated observation. In future, it has to be explored how to add additional provenance information about providers and users of the (aggregated) observations.

Our approach of aggregation in Linked Data also allows to utilize the semantics of the links and the objects. First of all, the changes to links as described in Section 3.2 can be translated into rules to check whether adding or removing links is allowed or not. In a next step, the process of adding and removing links during aggregation of observations might be automatized. Furthermore, the semantic reasoning can be used to decide, whether a certain aggregation procedure can be applied to a certain set of observations. Considering, e.g., a set of water level measurements along rivers in Germany, these should not be interpolated to Germany and the semantics can be used to recommend appropriate or disallow inappropriate aggregation processes. However, in order to realize such a system, an ontology of aggregation processes is needed which we consider to be work done in a longer time frame. We hope that our approach as presented in this paper will contribute towards such a semantically-enabled aggregation system.

Acknowledgments

The presented work is developed within the 52° North semantics community (http://52north.org/semantics), and partly funded by the European projects *UncertWeb* (FP7-248488, http://www.uncertweb.org/), *ENVIROFI* (FP7-284898, http://www.envirofi.eu/), *ENVISION* (FP7-249170, http://www.envision-project.eu/), and through the International Research Training Group on Semantic Integration of Geospatial Information funded by the DFG (German Research Foundation), GRK 1498. We are thankful for discussions with members of the Münster Semantic Interoperability Lab (*MUSIL*), the Spatial Data Infrastructures Unit of the Joint Research Centre (JRC) of the European Commission, and colleagues from the W3C Semantic Sensor Network Incubator Group.

References

1. Larssen, S., Sluyter, R., Helmis, C.: Criteria for EUROAIRNET - the EEA air quality monitoring and information network. Technical Report 12, European Environmental Agency (EEA) (1999)
2. Horalek, J., de Smet, P., de Leeuw, F., Conkova, M., Denby, B., Kurfuerst, P.: Methodological improvements on interpolating european air quality maps. etc/acc technical paper 2009/16. Technical report, EEA (2010)

3. Janowicz, K., Bröring, A., Stasch, C., Schade, S., Everding, T., Llaves, A.: A restful proxy and data model for linked sensor data. International Journal of Digital Earth (2011; accepted for publication)
4. Vega Lopez, I.F., Snodgrass, R.T., Moon, B.: Spatiotemporal aggregate computation: A survey. IEEE Trans. on Knowl. and Data Eng. **17**(2) (2005) 271–286
5. Bierkens, M., Finke, P., De Willingen, P.: Upscaling and Downscaling Methods for Environmental Research. Kluwer Academic Publishers (2000)
6. Stasch, C., Autermann, C., Foerster, T., Pebesma, E.: Towards a Spatiotemporal Aggregation Service in the Sensor Web. Poster Presentation. In: The 14th AGILE International Conference on Geographic Information Science. (2011)
7. Jeong, S.H., Fernandes, A., Paton, N.W., Griffiths, T.: A generic algorithmic framework for aggregation of spatio-temporal data. In: SSDBM '04: Proceedings of the 16th International Conference on Scientific and Statistical Database Management, Washington, DC, USA, IEEE Computer Society (2004) 245
8. Pebesma, E.J., de Kwaadsteniet, J.W.: Mapping groundwater quality in the netherlands. Journal of Hydrology **200**(1-4) (1997) 364 – 386
9. Journel, A., Huijbregts, C.: Mining geostatistics. Academic Press (1978)
10. Patni, P., Sahoo, S., Henson, C., Sheth, A.: Provenance aware linked sensor data. In Kärger, P., Olmedilla, D., Passant, A., Polleres, A., eds.: Proceedings of the Second Workshop on Trust and Privacy on the Social and Semantic Web. (2010)
11. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data – The Story So Far. International Journal on Semantic Web and Information Systems **5**(3) (2009) 1–22
12. Lassila, O., Swick, R.: Resource description framework (rdf) model and syntax specification. http://www.w3.org/tr/pr-rdf-syntax/. Technical report, W3C (1999)
13. Prud'hommeaux, E., Seaborne, A.: Sparql query language for rdf. w3c recommendation, http://www.w3.org/tr/rdf-sparql-query/. Technical report (2008)
14. Page, K., De Roure, D., Martinez, K., Sadler, J., Kit, O.: Linked sensor data: Restfully serving rdf and gml. In K., T., Ayyagari, A., De Roure, D., eds.: Proceedings of the 2nd International Workshop on Semantic Sensor Networks (SSN09). Volume Vol-522., CEUR (2009) 49–63
15. Phuoc, D.L., Hauswirth, M.: Linked open data in sensor data mashups. In Kerry Taylor, A.A.D.D.R., ed.: Proceedings of the 2nd International Workshop on Semantic Sensor Networks (SSN09). Volume Vol-522., CEUR (2009) 1–16
16. Patni, H., Henson, C., Sheth, A.: Linked sensor data. In: 2010 International Symposium on Collaborative Technologies and Systems, IEEE (2010) 362–370
17. Na, A., Priest, M.: OGC Implementation Specification 06-009r6: OpenGIS Sensor Observation Service (SOS). Open Geospatial Consortium (2007)
18. Janowicz, K., Compton, M.: The stimulus-sensor-observation ontology design pattern and its integration into the semantic sensor network ontology. In Ayyagari, A., Roure, D.D., Taylor, K., eds.: Proceedings of the 3rd International workshop on Semantic Sensor Networks 2010 (SSN10) in conjunction with the 9th International Semantic Web Conference (ISWC 2010). Volume 668., CEUR-WS (2010)
19. Zhao, J., Bizer, C., Gil, Y., Missier, P., Sahoo, S.: Provenance requirements for the next version of rdf. In: W3C Workshop RDF Next Steps. (2010)
20. Hartig, O., Zhao, J.: Publishing and consuming provenance metadata on the web of linked data. In: In: Proc. of 3rd Int. Provenance and Annotation Workshop. (2010)
21. Park, U., Heidemann, J.: Provenance in sensornet republishing. In: Proceedings of the 2nd International Provenance and Annotation Workshop, Salt Lake City, Utah, USA, Springer-Verlag (June 2008) 208–292

22. Liu, Y., Futrelle, J., Myers, J., Rodriguez, A., Kooper, R.: A provenance-aware virtual sensor system using the open provenance model. In: 2010 International Symposium on Collaborative Technologies and Systems (CTS). (2010) 330 –339
23. Lefort, L., Henson, C., Taylor, K., Barnaghi, P., Compton, M., Corcho, O., Castro, R., Graybeal, J., Herzog, A., Janowicz, K., Neuhaus, H., Nikolov, A., Page, K.: Semantic Sensor Network XG Final Report. W3C Incubator Group Report 28 June 2011. Available at: http://www.w3.org/2005/Incubator/ssn/XGR-ssn/. Technical report (2011)
24. Boose, E.R., Ellison, A.M., Osterweil, L.J., Clarke, L.A., Podorozhny, R., Hadley, J.L., Wise, A., Foster, D.R.: Ensuring reliable datasets for environmental models and forecasts. Ecological Informatics **2**(3) (2007) 237 – 247 Meta-information systems and ontologies. A Special Feature from the 5th International Conference on Ecological Informatics ISEI5, Santa Barbara, CA, Dec. 4-7, 2006 - Novel Concepts of Ecological Data Management S.I.
25. Schade, S., Craglia, M.: A future sensor web for the environment in europe. In: Proceedings of the 24th International Conference on Informatics for Environmental Protection - Enviroinfo2010. (2010)
26. Heuvelink, G.: Error Propagation in Environmental Modelling with GIS. Taylor & Francis (1998)
27. Pebesma, E., Cornford, D., Nativi, S., Stasch, C.: The uncertainty enabled model web (uncertweb). In: Environmental Information Systems and Services - Infrastructures and Platforms. (2010)
28. Bröring, A., Maué, P., Janowicz, K., Nüst, D., Malewski, C.: Semantically-enabled sensor plug & play for the sensor web. Sensors **11**(8) (2011) 7568–7605
29. Madden, S., Franklin, M.J., Hellerstein, J.M., Hong, W.: TAG: a Tiny AGgregation service for ad-hoc sensor networks. SIGOPS Operating Systems Review **36** (2001) 131–146
30. Williams, M. and Conford, D. and Bastin, L. and Pebesma, E.: OGC 08-122r2: Uncertainty Markup Language (UncertML) (2009)

Toward Situation Awareness for the Semantic Sensor Web: Complex Event Processing with Dynamic Linked Data Enrichment

Souleiman Hasan, Edward Curry, Mauricio Banduk, Sean O'Riain

Digital Enterprise Research Institute, National University of Ireland,
Galway, Ireland
{souleiman.hasan, ed.curry, mauricio.banduk,
sean.oriain}@deri.org

Abstract. Over the past few years there has been a proliferation in the use of sensors within different applications. The increase in the quantity of sensor data makes it difficult for end users to understand situations within the environments where the sensors are deployed. Thus, there is a need for situation assessment mechanisms upon the sensor networks to assist users to interpret sensor data when making decisions. However, one of the challenges to realize such a mechanism is the need to integrate real-time sensor readings with contextual data sources from legacy systems. This paper tackles the data enrichment problem for sensor data. It builds upon Linked Data principles as a valid basis for a unified enrichment infrastructure and proposes a dynamic enrichment approach that sees enrichment as a process driven by situations of interest. The approach is demonstrated through examples and a proof-of-concept prototype based on an energy management use case.

Keywords: situation awareness, semantic sensor networks, semantic web, linked data, dynamic enrichment, complex event processing, spreading activation, semantic similarity.

1 Introduction

The notion of Situation Awareness (SA) has emerged in two main fields: Information Fusion [1] and Human Computer Interaction [2]. The objective of situation awareness is to empower the user with an understanding of the developing relationships of interest between entities in question within a specific time and space [1]. SA techniques have been applied to improve user understanding within a range of systems, from the mission and safety critical role of helping pilots in the cockpit, to empowering business executives' with decision support to optimize business operations with real-time business intelligence [1].

As sensor networks deployments have increased, sensor information has become one of the main information flows within situation awareness systems. At the same time the introduction of web and semantic web technologies to sensor networks [3] has improved the accessibility of sensor data [4]. Enterprises are also finding more uses for sensors, from supporting the operational layers to the higher-level strategic decision making layers [5].

Sensor readings are usually limited in the amount of information they hold. The quality of SA is dependent on the quality of context available when the situation awareness is determined. Thus, there is a need to enrich sensor information flows with additional context from existing systems within the enterprise to conduct higher quality situational assessments.

In this paper we investigate the challenges associated with situation awareness in web sensor networks, we propose an approach to situational awareness utilizing a combination of Complex Event Processing (CEP) and Linked Data. In particular, we examine the validity of using linked data as a basis for sensor data enrichment. The approach utilizes dynamic enrichment over linked data streams, combined with CEP as the means to realize situation awareness in enterprises.

The contribution of this paper is the introduction of dynamic enrichment as a key enabler to realize situation awareness over large-scale and open web sensor networks. The paper proposes a model for dynamic enrichment based on spreading activation in linked data and the semantic similarity measures between information items and the situations of interest. It also proposes an evaluation framework for the approach.

The remainder of the paper goes as follows: *Section 2* motivates the need for a situation awareness mechanism for the web sensor networks along with some associated challenges. *Section 3* describes the proposed approach and details the dynamic enrichment process. *Section 4* demonstrates the approach via a prototype based on an energy management use case. *Section 5* summarizes briefly related work in situation awareness and enrichment. The paper concludes in *Section 6* with future directions.

2 The Need for Situation Awareness for Web Sensor Networks

Over the last few years there has been a proliferation in the use of sensors within different use cases, from air and water pollution monitoring, to machinery health monitoring within factories. The increased uptake is being driven by lower costs to buy and install sensors and the simplification of their deployment [4]. The indications are this trend is set to continue with the introduction of web-based open standards for sensor networks and the switch to open data licensing policies which will further increase the accessibility of sensor data [4].

Within business environments there is an increasing demand to support real-time decision making business process. When making a decision the value of information is higher and more useful for the decision makers when its freshness is higher [6]. This motivates the desire to expand the use of sensor networks upward within the knowledge and decision stacks of enterprises, from supporting technical low-level applications, to supporting higher-level decision making processes [5]. Nevertheless, users and organizations find it hard to interpret, understand and leverage the rapidly increasing quantity of information which necessitates the use of situation assessment mechanisms.

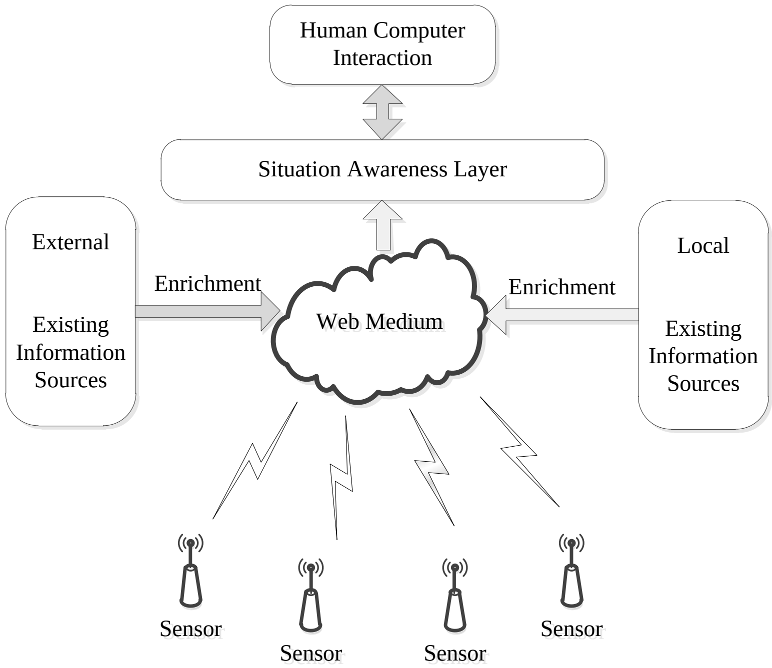


Fig. 1. A situation awareness layer positioned upon the sensor web

Situation awareness has been defined as "the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future" [7]. Elements of the environment include people, projects, devices, rooms, etc. Sensor networks can provide an enterprise with (near) real-time fresh flows of information items (i.e. readings, observations, events, etc.). The synergy between these dynamic information flows along with traditional data sources that contain rather static information increases the quality of the overall comprehension of relationships between elements in the enterprise (e.g. people, devices, rooms, products, etc.) and thus the quality of business status assessment.

In order to process their information flows, many enterprises employ systems that are dedicated to high rate information flow processing in addition to their traditional database management systems. Data Stream Management Systems (DSMS) and Complex Event Processing (CEP) systems have been adopted with commercial systems starting to appear in the last few years [8]. *Figure 1* illustrates how a situation awareness layer can be positioned upon web sensor networks to deliver higher-level insights.

2.1 Challenges with Situation Awareness

The process of creating situation awareness requires the configuration of the underlying information systems to process raw information flows and abstract them up to the level of situation awareness; refer to *example 1*. Within current state-of-the-art of DSMS and CEP systems, SA configuration is done by defining patterns of information flow items that are mapped to situations of interest for the target users [8]; refer to *code snippet 1* which shows an implementation of the scenario exposed in *example 1*.

Example 1.

Within an energy management scenario sensors observe the *kWh* energy usage of 12 heaters distributed among 3 floors in a building. Motion detection sensors are also in place to detect if a floor is empty. Typically, observing the energy consumption of devices and the emptiness of a floor does not provide in itself much value. That is because of the granularity level that might be non-useful for users who are not in the operational level and because these observations need to be linked together and drawn against other contextual information to make the result more actionable from an energy saving perspective. It would be better if after detecting that a floor was empty, the energy usage observations were aggregated over the devices in that floor for a time period (e.g. 30 minutes) and then compared with an acceptable threshold in order to conclude a more useful piece of information such as an excessive energy usage. That allows the users to move from a massive amount of data to higher level knowledge and facilitates the decision making with regard to energy saving.

To express the scenario explained in *example 1* in a pattern language such as the Event Processing Language (EPL) used in the open source complex event processing engine Esper [9], the following expression is used (simplified):

```
INSERT INTO ExcessiveEnergyUsageByFloor
SELECT a.floor as floor
FROM PATTERN [(a=FloorEmptySensor -> every
         b=DeviceEnergyUsageSensor(a.floor=b.floor))]
   .WIN:TIME(30 min)
GROUP BY a.floor
HAVING SUM(b.usage) > GetAcceptableThreshold(a.floor)
```

Code Snippet. 1. EPL implementation of the scenario explained in *example 1*

The following challenges can be identified along with the different activities needed for situation assessment:

Bridging the Information Gap.

One of the main challenges with defining SA is the need to bridge information gaps between different levels in an enterprise (e.g. operational to strategic) [5]. In technical terms this means defining patterns of interest in languages close to SQL (moving from `FloorEmptySensor` and `DeviceEnergyUsageSensor` to

`ExcessiveEnergyUsageByFloor` in *code snippet 1*), or sometimes, using user interfaces to help construct the patterns from known information flows and a controlled vocabulary [10]. This can become extremely challenging within open and large-scale environments, and even more difficult at web-scale. That is due to the large number of possible patterns and the large number of information flows and items' properties to be considered in patterns.

Heterogeneity of Information Flows.

Another difficulty results from the heterogeneous usage of ontologies, i.e. terminology or vocabulary, to publish semantic sensor data by different publishers. This complicates the task of the person responsible for defining the situations; the situation manager. It becomes very difficult to integrate terms from different publishers on a web-scale.

Uncertainty about Occurrence and Content of Information Flow Items.

Some real-world events might not be observed or vice-versa. Errors might also occur in the content of sensor readings. This results in a degree of uncertainty about what really happens in the real-world and affects the definition and evaluation of situations of interest. For instance, exact matching between situations of interest and observations could result in unfavorable false positives and false negatives.

Putting Information Flows into Context.

Sensor readings are usually limited in the amount of data that they contain (refer to *example 2*). This can be due to the limited resources of sensors and also the scope of the environment the sensor can observe. When used within an enterprise, sensor readings will often need to be interpreted within the context of other information systems including Enterprise Resource Planning (ERP), financial accounting systems, energy management systems, etc. Thus, the amount of data the item contains should be expanded in order to include information relevant to more situations of interest. This is a process known as data enrichment. Enriching sensor data adds further complexity as it can be difficult to define in advance and must be maintained during the system lifetime.

A more extensive discussion on challenges in situation assessment can be found in [11]. In the following we focus more on the enrichment issue.

3 Situation Awareness for Semantic Sensor Networks

In order to realize situation awareness for information flows from sensor networks and existing systems within the enterprise, we propose the use of a Complex Event Processing engine along with a dynamic enrichment component that enriches the information items before they can be considered for evaluation; refer to *figure 2*.

A loose control over the systems and information flows is assumed due to large-scale and openness motivated by the adoption of web technology. Thus, there is a need for a unified enrichment mechanism. The use of sensor networks that respect linked data principles [12] when publishing data forms a solid basis for enrichment. URIs can be used to refer to related entities in the enterprise and open linked data cloud. The sensor data can be enriched with useful information such as RDF data that is retrieved when dereferencing a URI; see *example 2*.

Beyond the concept of enrichment, we propose the idea of dynamic enrichment where the enrichment strategy is decided at run-time and depends mainly on the semantic similarity between the situations of interest registered in the system and the attributes of observed information items. Dynamic enrichment brings the following benefits:

- It simplifies the integration of context data into SA systems and thus simplifies the definition of situations of interest;
- Dynamism allows SA systems to quickly evolve;
- Semantic similarity reduces the gaps between different vocabularies used to describe items;
- Web data (external) sources can be easily included (weather data, partner information such as power mix of an electricity supplier, etc.)

Figure 2 illustrates the suggested approach to reach situation awareness in semantic sensor networks with more focus on the dynamic enrichment component. It is further explained in the following sub-sections.

3.1 Complex Event Processing

A Complex Event Processing [5] engine provides the processing model for evaluating situations of interest. After the situation of interest is expressed in the configuration of the CEP engine in the form of an event pattern, new information items can participate in the evaluation of the pattern if they are relevant. When a pattern is matched, a new higher-level event (e.g. `ExcessiveEnergyUsageByFloor` in *code snippet 1*) is generated and can participate in further processing or could be forwarded to an event consumer like a dashboard or a business process management tool.

3.2 Dynamic Enrichment with Linked Data

In order to address the challenge of defining and maintaining enrichment strategies for distributed and heterogeneous information flows, there is a need to support dynamic event enrichment. That means that enrichment is not defined during the design time of the system but left to the run-time where each information flow item is enriched according to different criteria; especially the situations of interest that are defined. *Figure 2* illustrates the main steps and factors that affect the proposed dynamic enrichment process. We will consider *examples 1* and *2* as well as *code snippet 1* while we are walking through the proposed approach.

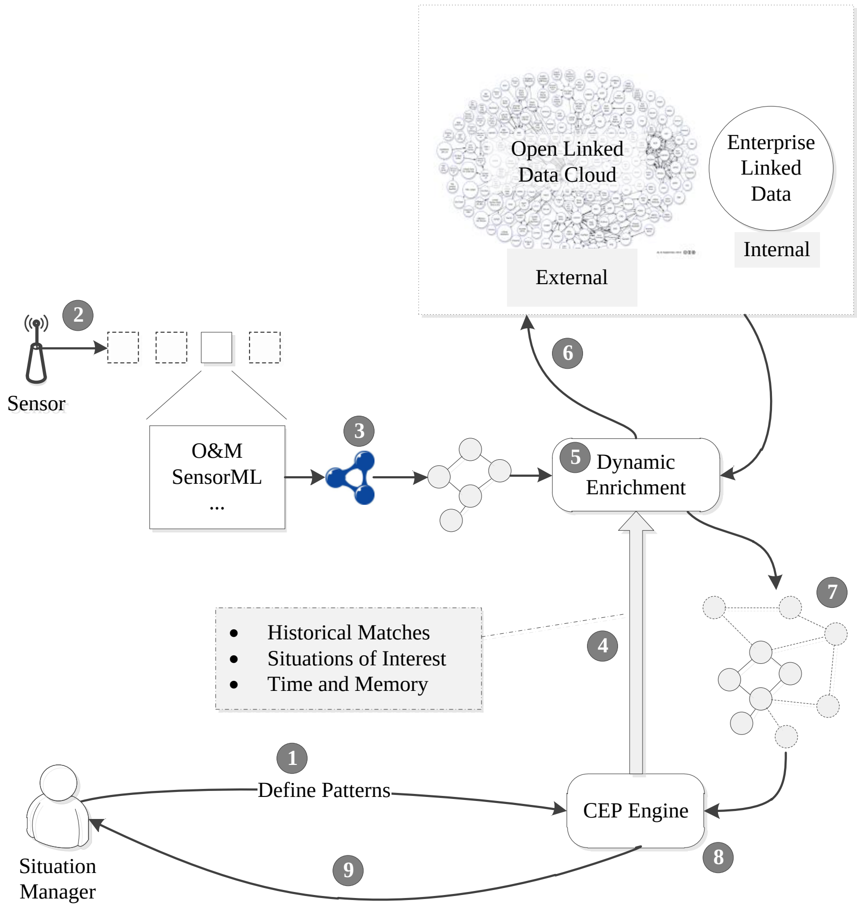


Fig. 2. Dynamic enrichment of Linked Sensor Data. The situation manager defines the situations of interest in the *CEP engine* (1), the sensor data are produced in *O&M* and *SensorML* formats during the run-time (2) then converted to RDF according to linked data principles (3), the *Dynamic Enrichment* component takes into consideration factors from the *CEP engine* such as historical matches, situations of interest, time and memory available (4) and decides on the data and time for enrichment (5), the enrichment is done by a spreading activation over the linked data graphs (6) and results in enriched sensor data (7) which is then evaluated against situations of interest (8), matches are forwarded to the end user (9).

Within this approach information items are adapted to linked data near the sensors with URIs referring to existing data entities in the enterprise or on the web of data. For example, the sensor readings of a heater' energy usage might come out of the sensors in an O&M XML format [13] containing the IP address of the sensor with the amount of energy usage. The linked data adapter converts these messages to an RDF format like N3 [14] and replaces the IP address by the appropriate URI of the heater in question. The resulting message would look like the one in *example 2*. More best practices about publishing linked sensor data can be found in the literature [15].

Example 2.

In *code snippet 1*, the `DeviceEnergyUsageSensor` reading may include just one RDF triple that describes the sensor observation about a specific device. The triple would use the URI of the device which would return more information about that device such as its type or the floor it is installed in when it is dereferenced. The sensor reading triple would look like the following:

```
<http://energy.deri.ie/resource/device/H008070>
<http://energy.deri.ie/ontology#usage> 55.6.
```

After the linked sensor readings reach the enrichment component, the component determines the information items, amount, and time for data enrichment. Enrichment itself is done by spreading activation [16] over the linked data graph starting from the content of information items. The direction and amount of spreading activation is guided by the semantic similarity between information items and the situations of interest. Spreading activation over linked data has been used for different purposes; see [17] as an example of spreading activation use for natural language querying over linked data. *Figure 3* shows an example of spreading activation for the `DeviceEnergyUsageSensor` reading.

We propose the following criteria as a basis for the enrichment decision:

- A semantic similarity measure between the information item content and the potential situation patterns that the information item can participate in. For example, the situation of interest in *example 1* is concerned with the accumulated energy usage of heaters that are installed in a *floor*. The sensor reading does not have data about the floor where the heater is installed (*example 2*). Semantic similarity is then used to guide the spreading activation process until satisfactory information about the heater’s floor is found. That might take one dereferencibility step for the URI `<http://energy.deri.ie/resource/device/H008070>` to find a predicate `<http://rooms.deri.ie/ontology#installed>` that leads to a resource of type `<http://rooms.deri.ie/ontology#Floor>`;
- The amount of time and memory available for the CEP engine to meet the user need to deliver the situation awareness in time. In order to improve performance effective caching is important. For example when we get *N* readings about the energy consumption of the same device, the device URI should be dereferenced in the first time and the result kept in the cache for the following times. The lifetime of an item in the cache should depend on a probabilistic or stochastic model that predicts the occurrence of events in the future;

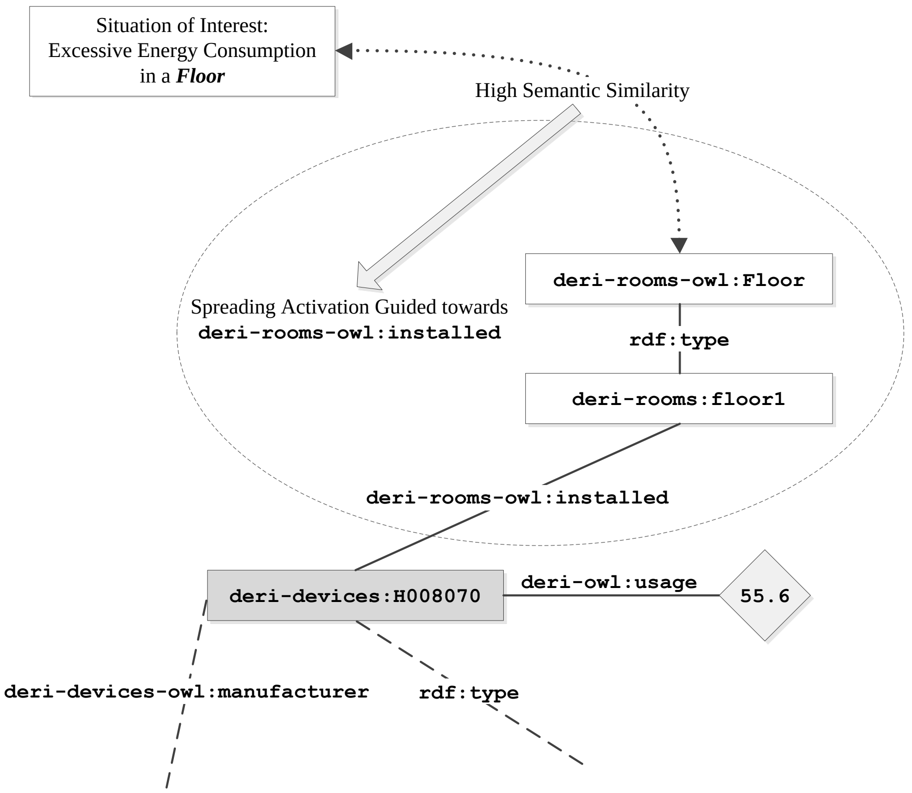


Fig. 3. Spreading activation for the `DeviceEnergyUsageSensor` reading

- The knowledge about useful previous enrichment or non-useful previous enrichment from the perspective of matched situations. For example, if the message in example 2 was enriched with the manufacturer of the device but it has never been used for matching, so there is no need to enrich with it the next time.

Table 1 summarizes the relationship between different criteria and the decision of enrichment.

Table 1. Criteria affecting different dimensions of the enrichment in the proposed approach

	Items for Enrichment	Direction of Enrichment	Amount of Enrichment	Time of Enrichment
Semantic Similarity	Yes	Yes	-	-
Response Time to conduct SA	Yes	-	Yes	Yes
Available Memory for CEP engine	Yes	-	Yes	Yes
Previous Matches	Yes	-	-	Yes

4 Proof of Concept: Energy Management Use Case

In order to support the argument made throughout this paper, a proof-of-concept prototype has been developed based on an enterprise energy management use case. The use case builds on the examples covered in the previous sections. This section briefly covers the technicalities and experience while implementing the scenarios.

In a typical modern office building there are many sources of power consumption such as Heating, Ventilation and Air Conditioning (HVAC) systems, lights and electronic devices. Tracking the operation of these systems can help in identifying information related to energy leaks and non-ecological actions. This information can be utilized to achieve reductions in energy consumption and cost saving. The purpose of a building energy management system is to gather data related to energy consumption and to present it in an actionable manner where actionable implies minimal effort to move from the presented knowledge to energy-related decisions.

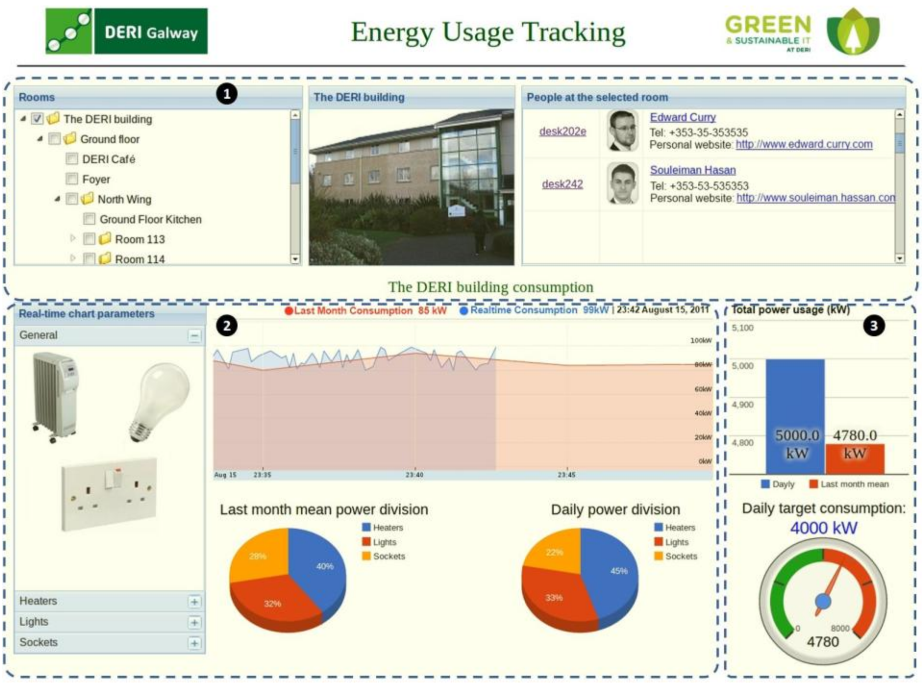


Fig. 4. A screenshot of the system dashboard. In (1) reference data from the enterprise linked data cloud can be seen; it is used for enrichment in the scenarios, in (2) instant measures by the sensors are shown and in (3) situation awareness is achieved by comparing the accumulative consumption with historical usage data and usage targets to detect high usage situations.

The system is deployed in the DERI office building. The information passing through the system is produced by 31 fixed energy consumption sensors covering office space, café, data centre, kitchens, conference and meeting rooms, computing museum along with 5 mobile sensors for devices, light and heaters' energy consumption as well as motion detection. Observations are collected by the sensor controller which triggers a broadcast of the information received. The sensor readings are adapted to RDF using the Jena framework [18] and enriched based on the enterprise linked data cloud that exists in DERI, which was developed in a previous project (see [19] for more information about Sustainable DERI project). The data is then sent to the CEP Engine. The CEP Engine makes situation assessment based on the pre-defined patterns of interest and once new data is generated by the engine it is forwarded to the user interface; refer to *figure 5* as an example screenshot.

To put the proposed approach into practice, basic energy usage sensor readings are sent without appropriate context information, such as in which floor or room of the building the consuming device is installed. A set of patterns of interest that aggregate energy usage according to the floors and rooms are registered in the CEP engine. The dynamic enrichment component does the necessary enrichment to include the missing pieces of information and allow the readings to be included in the evaluation of the deployed patterns. The system works as expected but a systematic evaluation is underway to evaluate the approach; see *Section 6*.

The CEP engine was extended to accept linked data events. Nevertheless, the core processing model is still a relational query model. This issue has not been investigated yet as we are more concerned with the enrichment part not with the matching functionality. However we believe that a deeper change in the processing model of the CEP engines is needed in order to effectively process Linked Sensor Data. We think that extending CEP with a more relaxed and approximate matching that is based on information retrieval approaches is more suitable for web deployments [20].

5 Related Work

Situation assessment has been identified as a key function in the Joint Directors of Laboratories (JDL) data fusion model [21]. It has been approached by different techniques ranging from probabilistic [22] to rule-based approaches [23]. Complex Event Processing (CEP) is a rule-based tool for processing dynamic information flows to help in situation assessment.

Sensor networks started to adopt semantic web technology in response to large-scale and heterogeneous deployments [3]. As a result, there has been a need to adapt CEP in order to process the semantic sensor web data [24]. Recently, situation awareness has been identified as one of the key challenges for semantic sensor networks [25]. Some works suggest the use of logic-based reasoners over RDF streams [24] but challenges such as performance and handling of uncertainty exist with such approaches in real-world scenarios [11];

Enrichment for information flows has been considered as a typical pattern in Message-Oriented Middleware (MOM) [26]. However, it has been considered as an external task used along with channel bandwidth considerations. We are not aware of research work that tackles the enrichment problem as a standalone problem in itself. However, the problem has been recognized in the event processing community as a main future research challenge [27].

6 Conclusions and Future Work

This paper discussed the synergy between information coming from semantic sensor networks together with existing information sources in enterprises to achieve high quality situational awareness to support decision making process. We argue the need for dynamic enrichment of information flows as a practical approach in large-scale and open systems. We also show how semantic sensor networks that respect linked data principles form a valid basis for dynamic and unified enrichment. We demonstrated a proof-of-concept prototype from the energy management world.

Future work would include the evaluation of the dynamic enrichment approach. Evaluation will be conducted towards: fewer amounts of memory usage and short time for enrichment as well as high precision and recall measures of matched situations. While the current work is concerned with a generic extension of CEP engines to do the enrichment, another future direction will examine the processing models of the CEP engines in order to realize natural language and approximate matching of situations over semantic sensor data.

7 Acknowledgements

This work has been funded by Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Lion-2).

8 References

1. Waltz, E., Llinas, J.: Multisensor data fusion. Artech House (1990).
2. Endsley, M.R., Garland, D.J.: Situation awareness: analysis and measurement. CRC (2000).
3. Lefort, L., Henson, C., Taylor, K., Barnaghi, P., Compton, M., Corcho, O., Garcia-Castro, R., Graybeal, J., Herzog, A., Janowicz, K., Neuhaus, H., Nikolov, A., and Page, K.: Semantic Sensor Network XG Final Report, W3C Incubator Group Report (2011). Available at http://www.w3.org/2005/Incubator/ssn/XGR-ssn/.
4. Sheth, A., Henson, C., Sahoo, S.S.: Semantic sensor web. IEEE Internet Computing. 12, 78-83 (2008).
5. Luckham, D.: The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems. Addison-Wesley Professional (2002).
6. Hackathorn, R.: Current practices in active data warehousing. DMReview. (2002).
7. Endsley, M.R.: Design and evaluation for situation awareness enhancement. In: Human Factors and Ergonomics Society Annual Meeting. pp. 97-101 (1988).
8. Cugola, G., Margara, A.: Processing flows of information: From data stream to complex event processing. ACM Computing Surveys Journal. (2011).
9. Esper Complex Event Processing Engine, http://esper.codehaus.org/.
10. Etzion, O., Niblett, P.: Event Processing in Action. Manning Publications Co. (2010).
11. Blasch, E., Kadar, I., Salerno, J., Kokar, M.M., Das, S., Powell, G.M., Corkill, D.D., Ruspini, E.H.: Issues and challenges in situation assessment (level 2 fusion). Journal of Advances in Information Fusion. 1, 122-139 (2006).

12. Linked Data: Design Issues, http://www.w3.org/DesignIssues/LinkedData.html.
13. Botts, M., Percivall, G., Reed, C., Davidson, J.: OGC sensor web enablement: Overview and high level architecture. GeoSensor networks. 4540, 175-190 (2008).
14. Notation 3, http://www.w3.org/DesignIssues/Notation3.
15. Barnaghi, P., Presser, M., Moessner, K.: Publishing Linked Sensor Data. In: The 3rd International Workshop on Semantic Sensor Networks (2010).
16. Collins, A.M., Loftus, E.F.: A spreading-activation theory of semantic processing. Psychological review. 82, 407 (1975).
17. Freitas, A., Oliveira, J., O'Riain, S., Curry, E., da Silva, J.: Querying Linked Data Using Semantic Relatedness: A Vocabulary Independent Approach. In: Natural Language Processing and Information Systems. 40-51 (2011).
18. McBride, B.: Jena: A semantic web toolkit. Internet Computing, IEEE. 6, 55-59 (2002).
19. Curry, E., Hasan, S., Hassan, U. ul, Herstand, M., O'Riain, S.: An Entity-Centric Approach to Green Information Systems. In: The 19th European Conference on Information Systems (ECIS) (2011).
20. Curry, E., Chambers, D., Lyons, G.: Extending message-oriented middleware using interception. In: Third International Workshop on Distributed Event-Based Systems. p. 32 (2004).
21. Steinberg, A.N., Christopher L. Bowman: Revisions to the JDL Data Fusion Model. Handbook of Multisensor Data Fusion (2001).
22. Das, S., Grey, R., Gonsalves, P.: Situation assessment via Bayesian belief networks. Information Fusion, 2002. In: The Fifth International Conference on. pp. 664-671 (2002).
23. Matheus, C.J., Baclawski, K.P., Kokar, M.M.: Derivation of ontological relations using formal methods in a situation awareness scenario. In: Proceedings of SPIE. pp. 298-309. Citeseer (2003).
24. Taylor, K., Leidinger, L.: Ontology-Driven Complex Event Processing in Heterogeneous Sensor Networks. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., and Pan, J. (eds.) The Semanic Web: Research and Applications. pp. 285-299. Springer Berlin / Heidelberg (2011).
25. Taylor, K., Ayyagari, A., De Roure, D.: Research Topics in Semantic Sensor Networks. Proceedings of the Semantic Sensor Network Workshop, p. Preface (2006).
26. Hohpe, G., Woolf, B., Brown, K.: Enterprise integration patterns. Addison-Wesley (2004).
27. Hinze, A., Sachs, K., Buchmann, A.: Event-based applications and enabling technologies. In: The Third ACM International Conference on Distributed Event-Based Systems - DEBS'09. 1 (2009).

Short Paper: Using SOAR as a semantic support component for Sensor Web Enablement

Ehm Kannegieser , Sandro Leuchter

Fraunhofer IOSB, Department of Interoperability and Assistance Systems
{ehm.kannegieser, sandro.leuchter}@iosb.fraunhofer.de

Abstract. Semantic service discovery is necessary to facilitate the potential of service providers (many sensors, different characteristics) to change the sensor configuration in a generic surveillance application without modifications to the application's business logic. To combine efficiency and flexibility, semantic annotation of sensors and semantic aware match making components are needed. This short paper gives the reader an understandig of the SOAR component for semantic SWE support and rule based sensor selection.

Keywords: SOAR, SCA, rule based sensor selection, service discovery

1 Introduction

It is a common conception that semantic service discovery is necessary to facilitate the Internet of Services because the plethora of potential service providers has to be matched to the specific needs of a service consuming value chain. From our point of view this is also true for the Internet of Things – in our case a sensor web – because there are many sensors with different characteristics available. We want to be able to change the sensor asset configuration in a generic surveillance application without modifications to the application's business logic. We also envision a SOA-like, wide area enterprise architecture for sensor webs where different sensor service providers will be combined in a cost efficient and flexible way. To achieve this, semantic annotation of sensors and semantic aware match making components are needed. In the remainder of this paper we describe our solution to this problem definition, based upon Sensor Web Enablement (SWE), Service Component Architecture (SCA) and the SOAR rule engine as well as a prototype application in the surveillance domain.

2 Sensor Web Enablement

2.1 Semantic Support for SWE

SWE is a suite of OGC standards, e.g. Sensor Markup Language (SensorML), Sensor Planning Service (SPS) and Sensor Observation Service (SOS), which provides an open interface for sensor web applications as described in [1].

In a time of rapidly developing semantic web, sensors and sensor data have to be accessible in a feasible kind of way, thus have their capabilities described semantically. Ontological descriptions and annotations achieve this by adding helpful metadata to sensors and data, harnessing the massive amount of available information.

Originating from a semantic aware service discovery that fits into SOAs and is based upon the SOAR rule engine, we have developed a semantic component to aid the process of sensor tasking and sensor data retrieval in a SWE environment by finding the most feasible sensors and related data.

In a proof of concept we introduce a perimeter control application with simple service enabled sensors. The component uses an ontological representation of attributes and capabilities of deployed sensors and a custom rule set that uses context information to deduce constraints of the current situation and proposes sensor services best suited to current task and context.

3 System Architecture

3.1 SOAR

State, Operator, Action, Result (SOAR) describes the solution of a problem as a number of state transitions. A starting state (representing the problem) is changed into a final state (representing the solution) by changes to the systems memory, done by rules. A SOAR rule consists of two parts: condition and action. The condition describes a specific working memory (WM) pattern. If this pattern is matched by changes (e.g. input) to the working memory, the action is triggered, changing WM into a new pattern which may trigger other rules. This way SOAR is able to react to system context changes and transit towards the desired system state.

There are three different ways to store knowledge in SOAR: the WM is an acyclic directed graph which represents all known facts. Rules that are changing WM are stored in the production memory and last but not least, preference memory (PM) maintains a ranking of operator feasibility. In case that there is more than one feasible operator, an impasse arises and – if solved – one or more production rules (chunks) are created in the production memory to avoid same (or similar) WM constellation in the future. This mechanism enables SOAR with basic learning capabilities [2; 3].

To help with useful recommendations, SOAR needs to rely on facts. Thus, knowledge of the subject matter (e.g. sensors and their feasibility in specific environmental conditions) is modeled in an ontology. This allows for a reliable initialization of WM with facts from ontology classes and derivation of rules from relations (e.g. “is better/worse than”, “is part”), if specified.

3.2 Service-Oriented Architecture

For reasons of flexibility and re-usability we chose the Open SOA Collaboration (OSOA) Service Component Architecture (SCA) to encapsulate the SOAR engine. SCA applications consist of composites which again consist of components (Figure 1) which can be implemented in different programming languages (e.g. Java). Apart from the specific implementation used, the behavior of the components is described to combine them into complex applications [6].

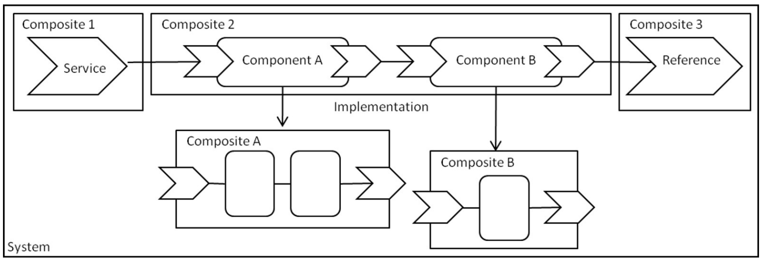


Figure 1. SCA-architecture

The component provides an interface for operation and communication with the encapsulating system. Through this interface the SOAR-kernel and its memories can be accessed (e.g. updating WM with new facts). The interface provided by the component is implemented in Java and features a number of methods for communication and control of the SOAR-kernel (see Table 1).

Table 1. SOAR-component methods

Method name	Performed action
start()/void stop()	Starts/Stops the SOAR-kernel
query(IDs)	Sends a query to the SOAR-kernel. A list of sensor ids (IDs) is passed to the method. Returns a list of sensor ids as result.
setContext(KVPs)	Writes updates/changes to the WM. A list of key-value-pairs is passed to the method. This is the most important method because it allows the WM to be changed. Changes to the WM may trigger rules that affect the recommendation of sensors.
setOntologyFile(String,String)	Sets the ontology and the corresponding XSLT file for transformation of semantic information into the SOAR-kernel. Two filenames are passed to the method.
setProductionsFile(String)	Sets the productions file which contains the production and preference rules needed for

sensor recommendation.

3.3 Ontology

While SensorML features description of physical sensor capabilities and quality, more general properties like cost or access restrictions plus the system context (e.g. weather conditions) which the sensors are matched with, needs to be described semantically. To model this, we chose the W3C standard OWL. XML encoded files can easily be transformed into arbitrary output formats using XSLT and XPATH [5] therefore easily be transferred into SOAR WM.

3.4 Case Study Evaluation

Objective of this study is to create a surveillance system that operates on an exchangeable set of sensors and adapt its recommendations to the changing environment conditions. A specific scenario is planned as follows: A defined area (e.g. small room, hallway, laboratory) is equipped with different sensors (e.g. photoelectric barriers, pressure contacts, acoustic, ultrasonic, video, and luminosity sensors) and actuators (e.g. horn, warning light) to detect movement and then sound an alarm. Therefore the SOAR-component is integrated into an Open Geospatial Consortium (OGC) SPS and SOS architecture which are described in length at [7; 8].

In the following, the workflow of a recommendation process is described: Firstly if there is a request, the SOAR-component is initialized with the current system context and sensor description by its setContext()-method. This method inserts a key value pair (attribute, value) which describes the sensors properties and relations into the WM structure. A list of (available) sensor ids corresponding to those sensor descriptions used for initialization is fetched from the SPS using the query()-method. The SOAR-component then elaborates a list of recommended sensors according to a set of rules specifying the optimization goal (e.g. cost, time, access) provided by the setProductionsFile()-method. The recommended sensor Ids are passed to the SOS for further use (e.g. for inserting observations from this sensors only).

4 Conclusion & Future Work

In this paper we have presented a semantic component capable of rule driven sensor selection. In a proof of concept a perimeter control application which uses a semantic representation of sensors to propose sensor services best suited to current task and context, was set up. Future work will focus on additional semantic capabilities of the SOAR-component such as inference and mapping ontological description of sensors, relations and system context, using additional semantic features of SOAR (e.g. semantic memory) and extending SWE support (e.g. sensor data interpretation). Further fields of activity will be semantic service discovery in the context of SOA, Semantic Grid and Cloud Computing.

5 References

1 Open Geospatial Consortium Inc.: Sensor Web Enablement, http://www.opengeospatial.org/ogc/markets-technologies/swe, 2009.
2 John E. Laird and Clare Bates Congdon: The Soar User's Manual, 2011.
3 The Soar Group (2009): Soar Website. University of Michigan. http://sitemaker.umich.edu/soar/home, 2009.
4 W3C (2004): OWL Web Ontology Language Overview. http://www.w3.org/TR/owl-features/, 2004.
5 W3C (o.J.): Extensible Markup Language (XML). http://www.w3.org/XML/, 2009.
6 Chappell, David: Introducing SCA. http://www.davidchappell.com/articles/Introducing_SCA.pdf, 2007.
7 Arthur Na; Mark Priest: Sensor Observation Service, 2008.
8 Johannes Echterhoff; Ingo Simonis: OGC 09-000. Sensor Planning Service, 2011.

Short paper: Enabling Lightweight Semantic Sensor Networks on Android Devices*

Mathieu d'Aquin, Andriy Nikolov, Enrico Motta

Knowledge Media Institute, The Open University, Milton Keynes, UK
{m.daquin, a.nikolov, e.motta}@open.ac.uk

Abstract. In this short paper, we present an architecture to deploy lightweight Semantic Sensor Networks easily based on widely available Android Devices. This approach essentially relies on deploying a SPARQL endpoint on the device, and federating queries to multiple devices to build Semantic Sensor Network applications.

1 Introduction

Research into semantic sensor networks has been focusing on the treatment and processing of data aggregated from large networks of sensors, often based on specialised equipments geographically distributed in large areas. [1] discusses a number of challenges related to Semantic Sensor Networks in such scenarios. The challenge we are particularly interested in relates to the ability for "rapid development of applications" that make use of Semantic Sensor Network. We especially look at applications in scenarios where it is needed to set-up networks of simple sensors quickly and easily (e.g., school projects, small experiments).

In recent years, the Android platform[1] became a de-facto standard for different types of mobile devices from several manufacturers. These devices possess several types of embedded sensors such as a camera, an accelerometer, a GPS sensor and a microphone. On the other hand, as shown by our previous work [2], the computational power of these devices already allows efficient processing of small to medium volumes of semantic data.

In this short paper, we describe a lightweight architecture to create small scale sensor networks based on Android devices, and an application that makes use of such a network of Android devices/sensors. To realise this, we adapt a triple store to be deployed on an Android device, which provides a shared repository populated through sensor-aware applications on the device. The information gathered in this shared repository is exposed through an externally accessible SPARQL endpoint, making it possible to build applications exploiting data collected from a network of devices, through query federation.

2 Overview

The idea on which this paper is based is very simple: exposing the sensors attached to an Android Device through a SPARQL endpoint and using SPARQL query federation so that the information gathered through these sensors can be exploited as the product of a Semantic Sensor Network (see Figure 1).

* Part of this research has been funded under the EC 7th Framework Programme, in the context of the SmartProducts project (231204).

[1] http://www.android.com/

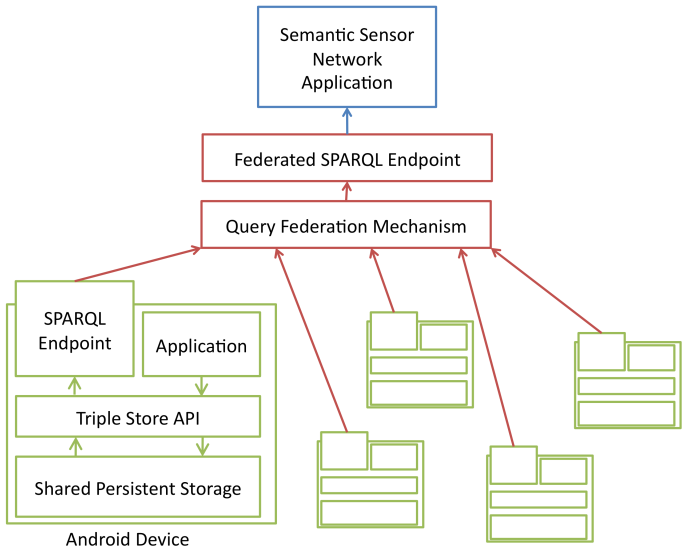


Fig. 1. Overview of the approach to create Semantic Sensor Networks out of Android devices.

In this approach, the use of the Semantic Sensor Network ontology is essential, as it allows to integrate the information coming from various devices and sensors. The idea here is that applications developed for the Android platform directly produce information using this ontology from sensors attached to the device. In this way, no post-processing is need for external applications to employ this information directly out of the SPARQL endpoint.

3 Implementation

At the core of our approach is the deployment of a triple store on the Android device, which is shared by the mobile applications populating it, using the Semantic Sensor Network ontology, and by the SPARQL endpoint deployed in the device. As discussed, in [2], Sesame[2] is, amongst the available options, the one that best fits an environment where only limited resources are available. We therefore adapted Sesame so that it can be deployable as an Android library. The Android environment is based on a specialised Java Virtual Machine, and Sesame being developed entirely in Java, most components of Sesame did not require any adaptation. Access to files is however different on the Android platform than it is on a usual computer. We therefore extended Sesame so that it provides a persistent RDF store using the shared, external storage available on most Android Devices (in SmartPhones, it corresponds to the SD card). In other terms, a shared persistent repository is installed on the Android device that is accessible, and can be populated, by applications accessing the device's sensors.

[2] http://www.openrdf.org/

The other element to be included on the Android device is a Web interface giving access, through the SPARQL protocol, to the content of the shared triple store. One of the difficulties here is to deploy a Web server on the such as device, being accessible externally. Luckily, the popular Web application server Jetty[3] has been ported to work on the Android Platform in iJetty[4], providing both a Web server and a servlet environment. We therefore implemented (a simplified version of) a SPARQL endpoint as a web application relying on our Android-adapted version of the Sesame API.

Finally, a mechanism is needed for the federation of SPARQL queries over multiple Android devices. In the cases where the data comes only from isolated and independent sensors, this federation mechanism can be very simple, as it only requires concatenating the results obtained from queries to different devices. In more complex scenarios where information from different sensors can be linked (such as discussed later), a more sophisticated mechanism is needed. We rely here on our own implementation of a distributed SPARQL query endpoint based on federating queries to multiple other SPARQL endpoints [3].

4 Example Application

To illustrate the benefits of the architecture we are proposing for lightweight Semantic Sensor Networks with Android, we developed a simple application used to collaboratively "map" a geographical area using pictures (for example, to give an idea of the views at certain points of a path, in an area that Google

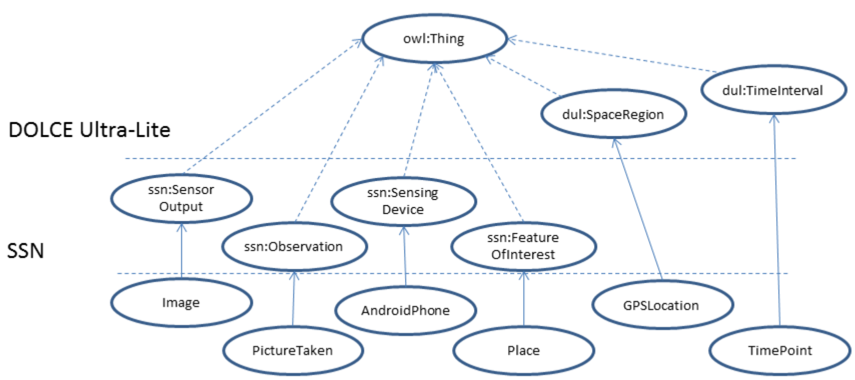


Fig. 2. Extension of the SSN ontology used in our example application.

We develop an application that can take pictures and represent the information about the picture and its location as an Observation using the extension of the Semantic Sensor Network ontology shown in Figure 2. This application records the path of the picture on the device, the location of the device at the time of taking it, as well as the time and the identifier of the device representing the sensor used to make the observation. We then used this application on several

[3] http://jetty.codehaus.org/jetty/
[4] http://code.google.com/p/i-jetty/

different SmartPhones. Using the simple SPARQL federation method described above, we implemented a Javascript application that displays the pictures taken from this network of phones on a map of the covered area (see Figure 3).

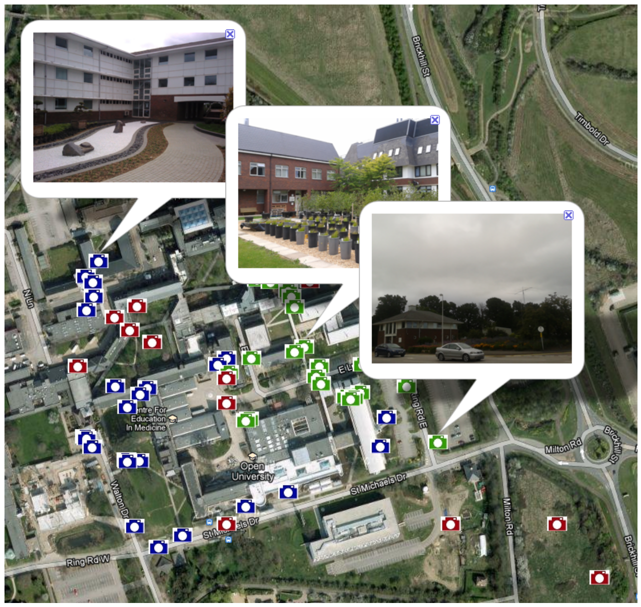


Fig. 3. Screenshot of the application mapping pictures taken from a network of Android phones using federated SPARQL queries.

5 Related Work

An architecture supporting a semantic sensor network commonly includes three layers: 1- Data layer: sensors producing observations; 2- Middleware layer: services responsible for sensor discovery and semantic data integration; 3- Service layer: services providing integrated data to the end-user applications. Implementing such an infrastructure for a specific use case requires dealing with five main challenges [1]:

- Choosing an appropriate *abstraction level* for sensor data representation.
- Adequate characterisation and management of the *quality of service*, including desired levels of data availability, completeness, and response time.
- Realizing *integration and fusion of data* from heterogeneous sensing devices.
- *Identification and location* of relevant sensor data sources.
- Supporting *rapid development of applications* handling integrated sensor data.

Depending on the requirements of the targeted use case scenarios, these issues can be addressed to a different extent, and a trade-off between them can be required. For example, ensuring high quality of service can make the architecture more complex and expensive and complicates deployment and application development.

Among existing solutions, the SemsorGrid4Env project[5] focuses on building large-scale semantic sensor networks for environmental management, in particular, for such critical scenarios as fire prevention and flood control. The proposed solution involves a multi-tier service-oriented architecture [4], which utilises a range of web services to integrate streaming data coming from sensor networks with data stored in static repositories. The nature of the scenarios requires the architecture to pay particular attention to such problems as quality of service, information latency, and security. The Sense2Web approach [5] uses the linked data principles to make sensor data publicly available via the Web. It allows the user to publish semantic descriptions of sensors and link them to other linked data resources (such as location URIs from DBPedia). The LinkedSensorData repository [6] implements a wrapper over meteorological data provided by sensors in the O&M XML format, combines the data in a single repository and makes it accessible with SPARQL queries. Similarly, the SensorMashup platform [7] assumes the existence of sensors producing streaming data and provides a semantic infrastructure for composing mashups over these data.

These architectures primarily concentrate on large-scale geographically distributed networks for industrial scenarios. We, on the other hand, look at more ad-hoc scenarios in which the deployment of a dedicated sensor network can be too complex and expensive. Other systems have been developed that provide RDF-based storing and querying of information on Android devices [8, 9]. They focus on the storage, querying and manipulation of RDF on the device, while we focus on how exposing information collected on the device through SPARQL can enable lightweight networks of devices as sensors.

6 Discussion

The architecture presented here is simple and lightweight by nature. It has however a number of advantages that favour the rapid development of applications relying on Semantic Sensor Networks, where the sensors are provided by one of the most available and affordable platforms. We presented an application that shows how somebody can easily set up a set of devices that act as a sensor network, providing information through a SPARQL interface for the application to integrate and use.

We can imagine many possible extensions to this application and other applications where a similar set-up could be considered, focusing on different types of sensor. For example, we could use the same application in a school trip dedicated to bird-watching. Groups of pupils would be given an Android phone where they could record with a picture seeing a particular type of bird. In this case, we could extend the application to also use the microphone of the phone to record the sound of the birds. Thinking about other sensors that could be used on an Android device, we could set-up a network of devices at different fixed positions in a building to record the vibrations on the floor of the building during the day, together with the sound intensity (for example, to find out about the impact of some building work). We could use a more complete combination of

[5] http://www.semsorgrid4env.eu/

sensors (camera, accelerometer, microphone) to check how busy different areas of a museum are during an opening day, and derive from this information the flow of visitors going from exhibitions to exhibitions.

In all these examples, the common aspect is that the sensor infrastructure is simple and lightweight. There is a need for an architecture that can be easily set-up, is highly re-usable, and is affordable. Also, the need for easy ways to semantically integrate data is very clear in such scenarios. In our application, we could annotate the pictures with the buildings they represent, and similarly, with the species of birds considered in the school trip scenario. Both the vibration and museum scenarios should be connected to a semantic model of the buildings being considered and to events happening during the day.

The approach using query federation makes such integrations easier. Indeed, an annotation service could be set up that provide an interface for users to annotate the pictures taken from the different phones with buildings or bird species. This service would not need to aggregate all the data in one place, but could simply use query federation to access information about the pictures and expose the annotations through its own SPARQL endpoint, therefore enriching the network with more information. Similarly, a SPARQL endpoint can be set-up that delivers data regarding the buildings and events in which a network is set-up. An interesting element is that such a SPARQL federation approach makes it easier to realise hierarchical networks: networks made of sub-networks. The use of the Semantic Sensor Ontology and of a coherent URI scheme would allow in this case to put together sensor networks that are heterogenous in nature and infrastructure.

References

1. Corcho, O., Garcia-Castro, R.: Five challenges for the semantic sensor web. Semantic Web Journal **1**(1) (2010)
2. d'Aquin, M., Nikolov, A., Motta, E.: How much semantic data on small devices? In: EKAW. (2010)
3. Miche, M., Erlenbusch, V., Allocca, C., Nikolov, A., Mascolo, J.E., Golenzer, J.: Final concept for storing, distributing, and maintaining proactive knowledge securely. Technical Report D4.1.3, SmartProducts Consortium (2011)
4. Gray, A.J.G., Garca-Castro, R., Kyzirakos, K., Karpathiotakis, M., Calbimonte, J.P., Page, K., Sadler, J., Frazer, A., Galpin, I., Fernandes, A., Paton, N., Corcho, O., Koubarakis, M., De Roure, D., Martinez, K., Gmez-Prez, A.: A semantically enabled service architecture for mashups over streaming and stored data. In: ESWC. (2011)
5. Barnaghi, P., Presser, M., Moessner, K.: Publishing linked sensor data. In: Semantic Sensor Networks, SSN. (2010)
6. Patni, H., Henson, C., Sheth, A.: Linked sensor data. In: Collaborative Technologies and Systems, CTS. (2010)
7. Le-Phuoc, D., Hauswirth, M.: Linked open data in sensor data mashups. In: Semantic Sensor Networks, SSN. (2009)
8. David, J., Euzenat, J.: Linked data from your pocket: The Android RDFContentProvider. In: Demo, ISWC. (2010)
9. Le-Phuoc, D., Parreira, J.X., Reynolds, V., Hauswirth, M.: Rdf on the go: An rdf storage and query processor for mobile devices. In: Demo, ISWC. (2010)

Short Paper: Annotating Microblog Posts with Sensor Data for Emergency Reporting Applications

David N. Crowley[1,2], Alexandre Passant[2], and John G. Breslin[1,2]

[1] School of Engineering and Informatics
National University of Ireland, Galway
firstname.lastname@nuigalway.ie
[2] Digital Enterprise Research Institute
National University of Ireland, Galway
firstname.lastname@deri.org

Abstract. The explosion in user-generated content (on the Social Web) published from mobile devices has seen microblog platforms like Twitter grow exponentially. Twitter is a microblogging platform founded in 2006, which by October 2010 had roughly 175m users and as of June 2011, Twitter processed 200m posts per day. Twitter data has been utilised to predict/report natural disasters, civil unrest, and media topics. Smartphones and other mobile devices contain an array of sensors but are under-utilised on the Social Web. In this paper, we propose a method for annotating microblog posts with multi-sensor data by representing it with ontologies such as SSN and SIOC. We present an alignment of these ontologies and outline an enhanced Twitter client that would allow users to enter an emergency mode where all or most of the available sensor data would be published as annotations to the users post, allowing relief organisations to use any data relevant.

Keywords: SSN, Microblog, SIOC, Citizen, Sensors, Social Sensing

1 Introduction

The unprecedented 96% growth in smartphone sales[1] and in user numbers on social platforms like Twitter (572,000 new accounts created on March 12, 2011) demonstrate the growth in the use of the mobile web. As microblogging lends itself to instantaneous updates, data related to events occurring around the world is created before it can be reported on by more traditional media methods or even by blog or blog-like services. In parallel with this growth in mobile-based microblogging, mobile devices themselves have begun to incorporate increasing amounts of sensors for various purposes, ranging from detecting light levels when a phone is placed close to one's head to accelerometers that can detect orientation changes and movement in various directions.

[1] http://www.gartner.com/it/page.jsp?id=1466313

In this short paper, we look at using microblogging platforms as citizen sensing/reporting platforms by adding mobile sensor data to user posts and describing that data using the SSN (Semantic Sensor Network)[2] ontology and the SIOC (Semantically-Interlinked Online Communities)[3] ontology. In particular, we outline applications for emergency scenarios, where people can report on events using microblogging while automatically attaching all available sensor data from their mobile devices (in order to provide context to emergency reports). The structure of this paper is as follows. Section 2 will describe related work in this area along with a brief review of mobile sensors. We will describe the Twitter Annotations initiative in Section 3, and how it can be used for sensor data annotations. Section 4 will detail the alignments required between the social and sensor data ontologies SIOC and SSN. Section 5 will outline our proposed 'emergency mode' microblogging client that allows users to upload all available sensor data with a post to aid relief workers/government agencies. We will present conclusions in Section 6.

2 Related Work

Sheth uses the example of Twitter posts during the Mumbai terrorist attacks in November 2008 when Twitter updates and Flickr feeds by citizens using mobile devices reported observations of these events in real time[1].[4] Twitter data has been used in event/disaster reporting and prediction[2]. Tapia *et al.* examined the usage of Twitter to aid relief workers with information regarding disasters, and they saw one method of using "microblogged data as ambient or contextual data to enrich the information provided to the NGO at the time of disaster"[3]. Mobile devices contain many sensor formats that provide information such as location (through GPS or cell tower locations) to create/add context to microblog posts and status updates. Companies like Foursquare use this contextual data to create various geo-social gaming/marketing applications.

In relation to microblog posts, at present GPS adds location to the data of the post made, but in the field of multi-sensor context awareness, researchers are currently examining ways to augment devices with an awareness of their situation and environment to add contextual understanding through the use of combined sensor data. As Gellersen *et al.* asserts "Position is a static environment and does not capture dynamic aspects of a situation"[4], and this concept can be applied to most single sensor data, but with multi-sensor context awareness the diverse sensor readings are combined and then with processing situational awareness can be derived. Situation awareness is the observation of surrounding elements/events in relation to the user, this perception of the immediate environment lets humans derive meaning and aids in decision making.[5]

[2] http://www.w3.org/2005/Incubator/ssn/XGR-ssn/
[3] http://sioc-project.org/
[4] http://www.telegraph.co.uk/news/worldnews/asia/india/3530640/Mumbai-attacks-Twitter-and-Flickr-used-to-break-news-Bombay-India.html
[5] http://en.wikipedia.org/wiki/Situation_awareness

Table 1. Mobile Sensing Types

Sensor Types	**Sensor Return Values**
Accelerometer	Acceleration along X, Y, Z axes (m/s^2)
Gyroscope	Angular speed along X, Y, Z axes measured in radians/second
Magnetic Field	Magnetic field in X, Y, Z axes measured in micro-Tesla μT
Orientation	Angle measurement along X, Y, Z axes in degrees
Proximity	Distance (cm)
NFC	A short-range wireless technology
GPS	Returns location if available (longitude and latitude)
Camera	Captures still images and video
Microphone	Allows for capture of audio
Compass	Standard directional compass values
Light	Light intensity in Lumens

Table 1 shows a non-exhaustive list of common mobile device sensors and their expected return values. Apple, Google, Nokia, and Microsoft have developed APIs in their mobile operating systems to allow access to these sensors, which allow developers to use sensor data in their applications. Sensor APIs allow application developers access to sensor readings to aid in user experience but also to allow for the collection of context data [5]. The structure for attaching sensor readings to microblog posts can take the format of Twitter annotations, which we will describe in the next section, or SSN-annotated SIOC posts for semantically-enhanced applications as we will describe later on.

3 Twitter Annotations

Twitter Annotations is an initiative from Twitter that allows additional structured metadata to be attached to tweets, going beyond the geotemporal annotations normally found in social media content. While the annotation or metadata is structured, it is open to the user or developer to decide what additional information is attached to the microblog post. There is an overall limit of 512 bytes for the metadata payload, but this may be expanded as usage increases.

As an example in JSON, data about a movie described in a tweet could be attached to the tweet using the annotation {“movie”:{“title”:“The Guard”}}, indicating that the title of the movie is “The Guard”. The guidelines for Twitter Annotations state that the goal is to “bring more structured data to tweets to allow for better discovery of data and richer interactions.”[6] In the sphere of citizen sensing, Twitter annotations can be seen as a way to standardise an emerging field of supplementing microblog posts with sensor data and, as with any area, standardisation is important. Figure 1 illustrates two examples of annotations in the Twitter Annotations JSON format. The first example describes a digital compass sensor in an Android mobile device that returns direction in degrees, and the second describes data returned from a three-axis accelerometer.

[6] `http://dev.twitter.com/pages/annotations_overview`

In this work, the Twitter Annotations format will be used for adding sensor/ multi-sensor data to tweets using the Twitter Annotations API, and will inspire how we attach sensor data (represented using the SSN ontology) to tweets, blog posts and other microblog posts described via SIOC.

Sample compass annotation (Android):
[{“AndroidCompass":{"DirectionDegrees":"83"}}]

Sample accelerometer annotation (Android) with values for X, Y, Z:
[{"AndroidAccel":{"XAcc":"0.00","YAcc":"3.00","ZAcc":"9.00"}}]

Fig. 1. Annotation Examples

4 Aligning the SIOC and SSN ontologies

SIOC allows the semantic interlinking of content items from forums, blogs and other social websites, and aims to enable the integration of online community information[6]. SIOC provides a Semantic Web ontology for representing rich data from the Social Web using the Resource Description Framework (RDF). By describing the social data contained within online communities (powered by blogs, wikis, and forums) using semantic technologies, SIOC enables this data to become a “Social Web of Data”[7].

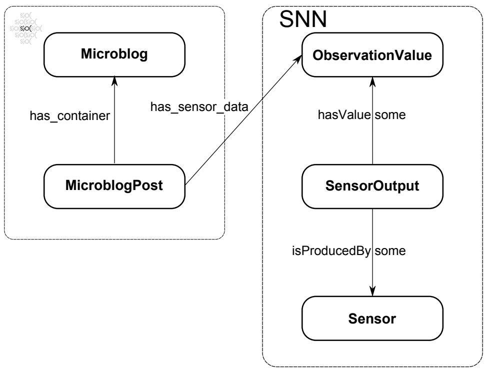


Fig. 2. Aligning the SSN Ontology with the SIOC Ontology

SIOC was originally written to describe web-based discussion on blogs and message boards, but with the SIOC Types module this has been expanded to include items like Microblog and MicroblogPost. SIOC has received significant adoption in commercial and open-source software applications[7]: it has been adopted in the core of Drupal 7 and around 100 applications use SIOC.

Figure 2 outlines our method for annotating microblog posts with sensor/multi-sensor data by representing it with ontologies such as SSN and SIOC. The property *has_sensor_data* will join *sioct:MicroblogPost* to *ssn:ObervationValue*. We proprose to create a SIOC Sensors (siocs) module to include this and future related properties. The sioct:MicroblogPost itself can have one or more ObservationValue(s). Figure 3 is an example of a microblog post with orientation sensor data attached. We define an *AndroidOrientation* sensor that has a defined *SensorOutput* that has value *OriObservationValue* a subclass of *ObservationValue* and has three properties *hasXQuantityValue*, *hasYQuantityValue*, and *hasZQuantityValue*, defined in a Citizen Sensors ontology (cs).

```
<sioct:MicroblogPost rdf:about="http://joebloggs.example.com/microblog/2622">
        <ssn:SensorOutput rdf:about="http://example.com/OriSensorOutput?sensor_id=1&amp;time=1313768104">
          <hasValue rdf:resource="http://example.com/OriObservationValue?sensor_id=1&amp;time=1313768104"/>
          <isProducedBy rdf:resource="http://example.com/AndroidOrientation?sensor_id=1"/>
        </ssn:SensorOutput>
        <ssn:Sensor rdf:about="http://example.com/AndroidOrientation?sensor_id=1"/>
        <sioc:content>Help!</sioc:content>
        <siocs:has_sensor_data>
                <cs:OriObservationValue rdf:about="http://example.com/OriObservationValue?sensor_id=1&amp;time=1313768104">
                  <cs:hasXQuantityValue rdf:datatype="&xsd;float">0.00</cs:hasXQuantityValue>
                  <cs:hasYQuantityValue rdf:datatype="&xsd;float">3.00</cs:hasYQuantityValue>
                  <cs:hasZQuantityValue rdf:datatype="&xsd;float">9.00</cs:hasZQuantityValue>
                </cs:OriObservationValue>
        </siocs:has_sensor_data>
</sioct:MicroblogPost>
```

Fig. 3. RDF Example: Android Orientation Sensor Annotation

5 Scenario

We will now describe a scenario whereby data from multiple sensors can be attached to microblog posts using the aforementioned alignments to aid in emergency scenarios. We are currently developing a semantic microblogging client for the Android platform that implements both Twitter Annotations and SSN-annotated SIOC posts for emergency reporting with sensor data.

In an emergency, the user could employ the semantic microblogging client and activate the emergency mode that would allow the application to annotate any available sensor data to their post (including photos). The available sensor readings could help emergency workers by attaching the direction the user is facing, noise levels in the surrounding area, light levels, direction of movement, and any other available data to the post. If GPS is unavailable, then from these sensors and the information extracted from the microblog post (place names or points of interest) an estimated location along a directional line could be calculated. In situations where a snapshot of data is not relevant, attaching aggregated

[7] `http://sioc-project.org/applications`

values/lists of values describing changes in activity, compass direction, and noise levels over time might better communicate the user's situation. Furthermore, the microblog post contents and the annotated sensor readings could aid emergency teams with reports including direction and lighting conditions.

6 Conclusion

By combining the Social Web and sensors, applications can provide an extension of social activities through sensors, as user activity is modelled by both voluntary user input and sensor data annotated to the posts. In this paper, we describe how this will be implemented using web standards like the SIOC onotology and by aligning SIOC with the SSN ontology to both describe users' posts semantically and attach contextual sensor data to the post through metadata annotations. We have described a scenario that uses this combined SIOC-SSN representation, based on a semantic microblogging client currently being developed for mobile devices that will enable emergency reporting functionality.

7 Acknowledgements

This work has been supported in part by Science Foundation Ireland under grant number SFI/08/CE/I1380 (Líon 2). We would like to thank Fabrizio Orlandi and Myriam Leggieri for their input.

References

1. Sheth, A.: Citizen Sensing, Social Signals, and Enriching Human Experience. Internet Computing, IEEE 13(4), 87–92 (2009)
2. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake Shakes Twitter Users: Real-Time Event Detection by Social Sensors, Proceedings of the 19th International Conference on World Wide Web, pp. 851–860. Raleigh, North Carolina, USA (2010)
3. Tapia, A.H., Bajpai, K., Jansen, B.J., Yen, J.: Seeking the Trustworthy Tweet: Can Microblogged Data Fit the Information Needs of Disaster Response and Humanitarian Relief Organizations, Proceedings of the 8th International ISCRAM Conference,. pp. 1-10, Lisbon, Portugal (2011)
4. Gellersen, H.W., Schmidt, A., Beigl, M.: Multi-Sensor Context-Awareness in Mobile Devices and Smart Artefacts. Mobile Networks and Applications 7(5), 341–351 (2002)
5. Lane, N.D., Miluzzo, E., Hong Lu, Peebles, D., Choudhury, T., Campbell, A.T.: A Survey of Mobile Phone Sensing. Mobile Networks and Applications 48(9), 140–150 (2010)
6. Berrueta, D., Brickley, D., Decker, S., Fernández S., Görn, C., Harth, A., Heath, T. , Idehen, T., Kjernsmo, K., Miles, A., Passant, A., Polleres, A., Polo, L., Sintek, M.: SIOC Core Ontology Specification. W3C Member Submission 12 June 2007, `http://www.w3.org/Submission/sioc-spec/`
7. Bojars, U., Passant, A., Cyganiak, R., Breslin, J.G.: Weaving SIOC into the Web of Linked Data, Proceedings of the WWW 2008 Workshop on Linked Data on the Web (LDOW2008), Beijing, China (2008)

Short Paper: Addressing the Challenges of Semantic Citizen-Sensing

David Corsar, Peter Edwards, Nagendra Velaga, John Nelson, and Jeff Pan

dot.rural Digital Economy Hub,
University of Aberdeen,
Aberdeen, UK.
dcorsar,p.edwards,n.r.velaga,j.d.nelson,jeff.z.pan}@abdn.ac.uk
http://www.dotrural.ac.uk

Abstract. The challenges of the sensor web have been well documented, and the use of appropriate semantic web technologies promises to offer potential solutions to some of these challenges (for example, how to represent sensor data, integrate it with other data sets, publish it, and reason with the data streams). To date a large amount of work in this area has focused on sensor networks based on"traditional" hardware sensors. In recent years, citizen sensing has became a relatively well-established approach for incorporating humans as sensors within a system. Often facilitated via some mobile platform, citizen sensing may incorporate observational data generated by hardware (e.g. a GPS device) or directly by the human observer. Such human observations can easily be imperfect (e.g. erroneous or fake), and sensor properties that would typically be used to detect and reason about such data, such as measurements of accuracy and sampling rate do not exist. In this paper we discuss our work as part of the Informed Rural Passenger project, in which the passengers themselves are our main source for transport related sensing (such as vehicle occupancy levels, available facilities). We discuss the challenges of incorporating and using such observational data in a real world system, and describe how we are using semantic web technologies, combined with models of provenance to address them.

Keywords: Citizen-Sensing, Semantic Sensing, Semantic Citizen-Sensing, Provenance, Semantic Web

1 Introduction

The challenges of the sensor web have been well documented in, for example, [21], [22] and [8]. Documented challenges include: modeling, querying, and reasoning with large scale sensor data [8, 11, 17, 15]; identification of, and integration with other relevant data sets, at scale [8, 11, 18, 24, 7]; characterizing and managing sensor data quality [8]; and supporting rapid application development [8].

The use of semantic web technologies offer potential solutions to some of these challenges. Ontologies, such as the W3C SSN XG ontology[1] provide mod-

[1] http://www.w3.org/2005/Incubator/ssn/XGR-ssn

els for sensors, sensor networks, and observations; and linked data [5] enables integration of sensor data with other data sets [4, 13, 18]. Sensors typically produce streams of data, and so there is potential for using technologies such as RDF stream querying [6, 3] (as explored in [15]) and RDF stream reasoners (e.g. [2, 23]) to support the use of that data. Further, Application Programming Interfaces (APIs), such as the Linked Data API[2] offer support for rapid application development.

To date a large amount of work in this area has focused on sensor networks based on "traditional" hardware sensors. In recent years, citizen sensing [19] has became a relatively well-established approach for incorporating humans as sensors within a system. Often facilitated via applications (apps) running on a mobile phone, citizen sensing may generate observational data by hardware (e.g. a GPS device) or directly by the human observer. Such human observations can easily be imperfect (e.g. erroneous, incomplete, or fraudulent), and so, as with any open system, this raises issues such as information quality (IQ) [10], reliability, trust, and reputation [14].

One further challenge of citizen sensing, is that for observations generated directly by the human observer, sensor properties that would typically be used to detect and reason about imperfect data, for example measurements of accuracy and sampling rate, do not exist. Similar problems exist with observations generated by the mobile phone's hardware: often the mobile APIs provide few details such as data sheets (describing sensor capabilities), settings used for observations, and, in some cases, which sensor generated an observation[3].

This lack of information makes it difficult to perform the necessary assessments of observations produced using citizen sensing. Semantic web technologies potentially have a role to play here by, for example, providing additional contextual information for use in assessment processes.

In this paper we describe an example real-world system which combines citizen sensing with semantic web technologies (section 2); discuss some of the challenges faced by this system (section 3); and describe how we are addressing those challenges (section 4).

2 Example System

As part of the Informed Rural Passenger (IRP) project[4], we are investigating the challenges of developing a trusted passenger information system (PIS) for rural areas. In our system the passengers themselves are our main source of transport related sensing, performed using a mobile app. The app enables passengers to contribute observations about their journey on public transport, including observations generated directly by the phone (e.g. location, presence of Wi-Fi) and by the passenger (e.g. occupancy level, and perceived vehicle temperature).

[2] http://code.google.com/p/linked-data-api/

[3] For example, the Apple iPhone iOS's location API uses either the cellular network, Wi-Fi, or GPS sensor to determine location, but does not indicate which was used.

[4] http://www.dotrural.ac.uk/irp

Using linked data principles, this data is then integrated with other relevant data sets, and used as the basis of a PIS, which provides passengers with details, including real-time bus locations, delays, and expected arrival times. This therefore gives the potential for any imperfect data passed as input to the system to adversely effect its outputs, reducing user trust in the system.

3 Challenges of Semantic Citizen Sensing

In developing the IRP PIS, we have identified a set of issues, which extend those defined for the sensor web, and, we believe, require to be addressed by any system which incorporates humans as a source of sensor data, in order to remain trusted by its users. These challenges are raised due to the potential generation of imperfect data by humans, and the lack of information for identifying and reasoning about it.

Challenge 1, is one of the most pressing: the need to characterise and manage constructs not just of data quality, but also of, for example, reliability, reputation, and trustworthiness, which can use the available types of data.

This gives rise to **challenge 2**: maximising the data available for making those assessments. Here, identifying and integrating the sensor data with appropriate external data sets can help address this challenge. Related to this are: **challenge 3**, selecting an appropriate model for describing the citizen sensors and their observations, the possible granularity of which is limited by the lack of information about them; and **challenge 4**, integrating the qualitative observations generated by humans with the machine generated quantitive observations.

In real-time information systems, short response times are vital; however, processes such as data integration and data assessments potentially conflict with this requirement. Further, the additional data generated by these processes adds to the amount that must be stored and processed. This gives rise to **challenge 5**: designing a system architecture which uses an appropriate combination of technologies (e.g. for storing and reasoning about the data), which enable the system to perform well while maintaining an acceptable response time.

Finally, **challenge 6** relates to ensuring user privacy, especially when sensitive data such as location is being collected and used as the basis of information passed to other users and/or services. Addressing this challenge is made more difficult by the integration with other data sets, which potentially provide additional data which can be used to violate a user's privacy.

4 Addressing These Challenges

Within the IRP project we are addressing the above challenges by, firstly exploring the data available within the application domain, and secondly investigating how it can be integrated to form an information ecosystem supporting a range of services which perform PIS functions and data assessments. Whilst the solutions below are outlined within the context of IRP, we believe they are generalisable to other systems incorporating humans as sensors.

The information ecosystem that we are developing to address challenge 2 integrates the passenger observations with various other types of data, including user profiles, social networks, and various types of transport information, such as: operator timetables; NaPTAN and NPTG[5] datasets[6]; road maps; and details of roadworks and travel disruptions[7], which themselves link to NPTG. We will also reverse geocode location points obtained from passengers and, where possible, link them to the nearest road/railway line and nearest settlement in the NPTG.

Fig. 1 outlines how we are addressing challenges 3 and 4, by basing our model of citizen sensors and observations on the SSN XG ontology, with domain extensions describing: mobile phones as sensor platforms, people as sensors, and the observations generated via our mobile app (Trips represent timetabled journeys).

Within the ecosystem, we also maintain a provenance record of the data. Provenance has previously been identified as essential in supporting reliability, discovery, trust [20], and quality assessment [10] of online information, and so may play an important role in assessments of data in the ecosystem. Fig. 1 shows how we use the Open Provenance Model Vocabulary[8] (OPMV) encoding of OPM [12] to provide explicit provenance information about observations. OPM defines provenance using causal relationships between *Artifacts*, *Processes*, and *Agents*, which we use, for example, to link observations generated by a phone's hardware to the passenger controlling the sensing.

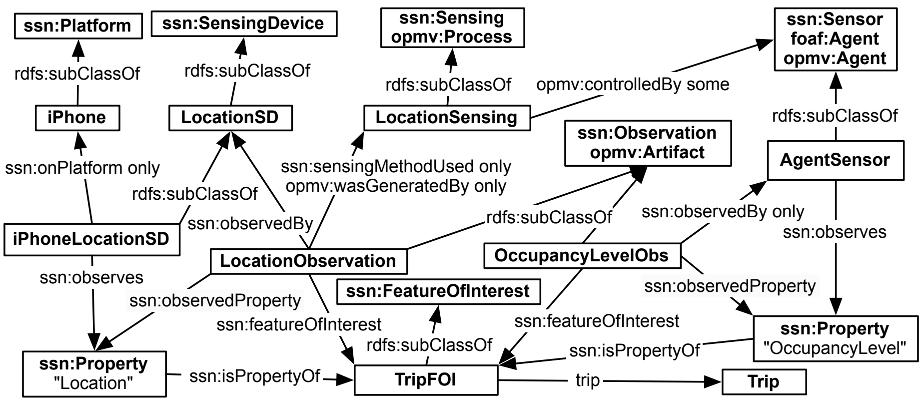


Fig. 1. SSN XG ontology extended for citizen sensing and provenance for IRP.

One reason for maintaining this information within the ecosystem is to support various types of data assessment, particularly of IQ and trustworthiness.

[5] http://data.gov.uk/linked-data
[6] NaPTAN provides details of all UK access points to public transport; NPTG provides details of all UK settlements and roads connected to the public transport network.
[7] Provided by http://trafficscotland.dataincubator.org/
[8] http://purl.org/net/opmv/ns-20101006

IQ assessments of data typically analyse various dimensions of the data, and so the additional information should be beneficial; for example, other members of our research team are currently investigating the role of provenance in IQ assessments of linked sensor data [1]. The multi-agent community have extensively studied models of trust and reputation [16, 14], which often rely on analysing past interactions between agents (i.e. analysing the provenance of interactions), while others combine trust, provenance and social networks [9]. As part of addressing challenge 1, we are currently investigating how these models can be applied within the ecosystem.

We will incorporate any data assessments and their results within the ecosystem as part of the provenance record (as subclasses of OPMV Process and Artifact classes respectively). This will allow services/applications (including those making new assessments) to make use of these assessments if appropriate.

The nature of the IRP project requires that it handles large quantities of data and still functions reliably in real time. To help support this and address challenge 5, passenger contributed observations are currently stored in a database, and exposed as linked data using the D2R server[9]. This setup takes advantage of the strengths of databases (such as scaling to large data sets, and handling multiple concurrent read, update, and delete operations). However, the disadvantage is that it does not exploit many of the advantages of semantic web technologies, such as the ontology based querying and reasoning.

5 Conclusions and Future Work

In this paper we have identified a set of challenges, which we believe, require to be addressed by any system that incorporates humans as a source of sensor data. We propose the use of semantic web technologies to help address these challenges, and illustrate their use in the development of a real-time PIS for rural areas.

We currently have three strands of future work addressing challenges 1, 5, and 6: developing a trust model for the ecosystem; evaluating the performance of different options for storing and reasoning about streaming linked sensor data, to determine if a combination can be found that provides (some of) the advantages of semantic web technologies without negatively impacting overall performance; and investigating how we can ensure user privacy.

Acknowledgements The research described here is supported by the award made by the RCUK Digital Economy programme to the dot.rural Digital Economy Hub; award reference: EP/G066051/1

References

1. Baillie, C., Edwards, P., Pignotti, E.: Assessing Quality in the Web of Linked Sensor Data. In: Proc. of AAAI-11 (2011)

[9] http://www4.wiwiss.fu-berlin.de/bizer/d2r-server/

2. Barbieri, D., Braga, D., Ceri, S., Della Valle, E., Grossniklaus, M.: Incremental Reasoning on Streams and Rich Background Knowledge. In: The Semantic Web: Research and Applications. vol. 6088, pp. 1–15. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
3. Barbieri, D.F., Braga, D., Ceri, S., Della Valle, E., Grossniklaus, M.: C-SPARQL: SPARQL for continuous querying. In: Proc. of the WWW'09. pp. 1061–1062. WWW '09, ACM, New York, NY, USA (2009)
4. Barnaghi, P., Presser, M.: Publishing Linked Sensor Data. In: Taylor et al. [22]
5. Berners-Lee, T.: Linked Data. IJSWIS 4(2), 1 (2006)
6. Bolles, A., Grawunder, M., Jacobi, J.: Streaming SPARQL - Extending SPARQL to Process Data Streams, Lecture Notes in Computer Science, vol. 5021, chap. 34, pp. 448–462. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
7. Compton, M., Neuhaus, H., Taylor, K., Tran, K.N.: Reasoning about Sensors and Compositions. In: Taylor et al. [21], pp. 33–48
8. Corcho, O., García-Castro, R.: Five challenges for the Semantic Sensor Web. Semantic Web Interoperability, Usability, Applicability 1(1), 121–125 (Jan 2010)
9. Golbeck, J.: Combining provenance with trust in social networks for semantic web content filtering. In: Proc. of IPAW 2006. vol. 4145, pp. 101–108. Springer (2006)
10. Hartig, O., Zhao, J.: Using Web Data Provenance for Quality Assessment. In: Proc. of Workshop on Semantic Web and Provenance Management at ISWC (2009)
11. Kessler, C., Janowicz, K.: Linking Sensor Data - Why, to What, and How? In: Taylor et al. [22]
12. Moreau, L., Clifford, B., Freire, J., Futrelle, J., Gil, Y., Groth, P., Kwasnikowska, N., Miles, S., Missier, P., Myers, J., Plale, B., Simmhan, Y., Stephan, E., den Bussche, J.V.: The open provenance model core specification (v1.1). Future Generation Computer Systems (July 2010)
13. Page, K., De Roure, D., Martinez, K., Sadler, J., Kit, O.: Linked Sensor Data: RESTfully serving RDF and GML. In: Taylor et al. [21], pp. 49–63
14. Ramchurn, S.D., Huynh, T.D., Jennings, N.R.: Trust in Multiagent Systems. The Knowledge Engineering Review 19(1), 1–25 (2004)
15. Rodríguez, A., McGrath, R., Liu, Y., Myers, J.: Semantic Management of Streaming Datas. In: Taylor et al. [21], pp. 135–147
16. Sabater, J., Sierra, C.: Review on computational trust and reputation models. Artificial Intelligence Review 24, 33–60 (2005)
17. Sabou, M., Kantorovitch, J., Nikolov, A., Tokmakoff, A., Zhou, X., Motta, E.: Position Paper on Realizing Smart Products: Challenges for Semantic Web Technologies. In: Taylor et al. [21], pp. 135–147
18. Sequeda, J., Corcho, O.: Linked Stream Data: A Position Paper. In: Taylor et al. [21], pp. 148–157
19. Sheth, A.: Citizen Sensing, Social Signals, and Enriching Human Experience. IEEE Internet Computing 13(4), 87–92 (2009)
20. Simmhan, Y.L., Plale, B., Gannon, D.: A survey of data provenance in e-science. ACM SIGMOD Record 34(3), 31–36 (2005)
21. Taylor, K., Ayyagari, A., De Roure, D. (eds.): Proceedings of the 2nd International Workshop on Semantic Sensor Networks (SSN09) (2009)
22. Taylor, K., Ayyagari, A., De Roure, D. (eds.): Proceedings of the 3rd International Workshop on Semantic Sensor Networks (SSN10) (2010)
23. Thomas, E., Pan, J.Z., Ren, Y.: TrOWL: Tractable OWL 2 Reasoning Infrastructure. In: the Proc. of ESWC2010 (2010)
24. Tran, K.N., Compton, M., Wu, J., Gor; R.: Short Paper: Semantic Sensor Composition. In: Taylor et al. [22]

Demonstration: *Sensapp* — An Application Development Platform for OGC-based Sensor Services

Dumitru Roman, Xiaoxin Gao, and Arne-Jørgen Berre

SINTEF, Oslo, Norway
dumitru.roman@sintef.no, gao1738@qq.com, arne.j.berre@sintef.no

Abstract. This paper introduces the Sensapp platform, a semantic and OGC-based sensor application platform to enable users to register, annotate, search, visualize, and compose OGC-based sensors and services for creating added-value services and applications. Functionalities of Sensapp such as sensor registration, sensor data visualization, visual composition and generation of executable service compositions are presented through the demo.

Keywords: OGC services, service annotation, discovery, and composition, sensor Web application.

1 Introduction

With the dramatic increase of sensor devices, large scale management of real-time data from such devices has become a real issue. Abstracting, selecting, and presenting real-time sensor data to end-users and decision makers in a suitable manner is a key requirement for enabling better decision making when dealing with processes involving real-time sensor data. Moreover, the need for supporting application developers in making sense of the huge amounts of real-time sensor data and using the data for creation of added-value applications and services implies development of novel platforms enabling faster and smarter development of added-value services. *Sensapp* (short for "*Sens*or *ap*plication *p*latform") is being developed as a platform addressing such needs. Focusing on the use of open standards such as those developed by the Open Geospatial Consortium (OGC)[1] and World Wide Web Consortium (W3C),[2] Sensapp aims to deliver a semantic and OGC-based sensor Web application platform to enable users to register, search, visualize, and compose OGC-based sensors and services for creating added-value services and applications on the Web.

Figure 1 provides an overview of the platform and its main components. The major stakeholders/roles in a Sensapp environment are resource providers, app/service developers, and application consumers (typically decision makers). Resource providers provide different kinds of resources such as sensors and data and

[1] http://www.opengeospatial.org/
[2] http://www.w3.org/

processing services. Data formats and protocols for accessing such resources are usually proprietary. The app/service developer is the main stakeholder interacting directly with all Sensapp components. Through the registration facility, the app developer will package and provide the sensor, data, and processing services as standardized OGC interfaces (e.g., Sensor Observation Services (SOS) [1], Web Feature Services (WFS) [2], Web Processing Services (WPS) [3], Sensor Event Services (SES) [4], etc). These OGC service interfaces are then semantically annotated through the annotation functionality of the platform. The Resource Description Framework (RDF) [5] annotations are used in the discovery and composition components. The discovery functionality enables enhanced search for services, which in turn will be used in the composition process where new added-value services are created. Composition is done by the app developer in a visual manner, based on the Business Process Modeling Notation (BPMN) [6]. The composition component contains facilities for data mapping, where semantic annotations of services are used. Once a composition is created, an executable representation of the composition is generated in Web Service Business Process Execution Language (WS-BPEL) [7] and a service interface (typically in WSDL) is created for the newly developed service. Based on portlets technologies (in particular Java Portlet Specification [8]), the platform can generate graphical components (scenario websites) corresponding to the developed services. The end user (typically decision makers) can consume the added-value services through the generated scenario websites.

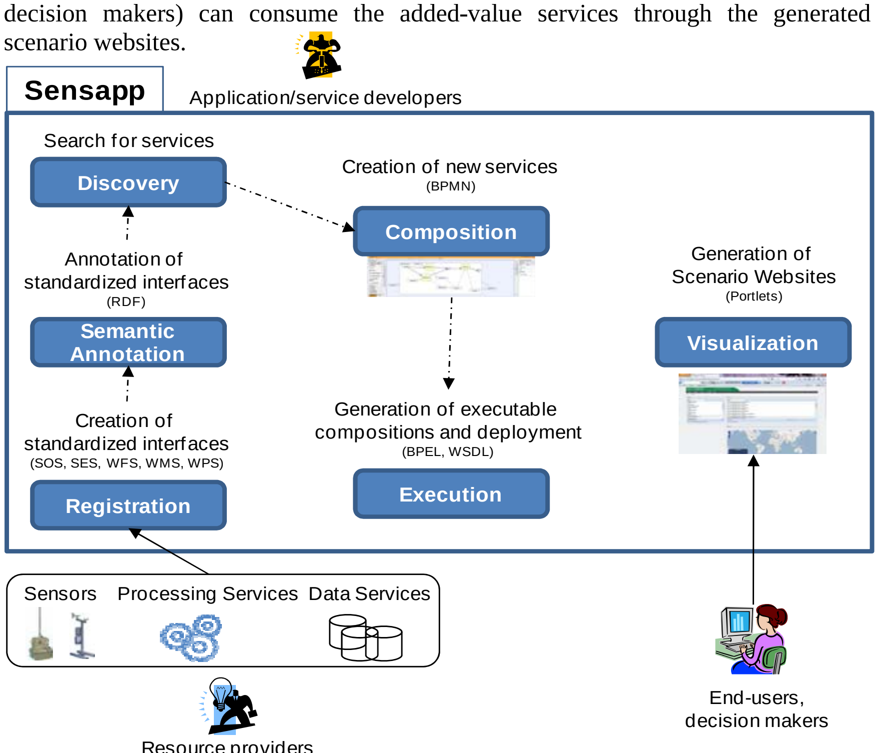


Fig. 1. Sensapp overview

By supporting abstraction of sensor data and services to standardized OGC interfaces/services, semantic annotation of such interfaces, enhanced discovery and composition of services, and data visualization on maps and charts, Sensapp aims to enable better access to sensor data and to create opportunities for faster and smarter development of added-value services based on real-time sensor data.

2 Demonstration

The demo will present some of the functionalities of Sensapp in particular related to sensor registration and visualization, visual composition and generation of executable compositions:

1. *Registration of OGC services*: Demonstrates how OGC services are registered to the Sensapp platform.
2. *Registration of individual sensors*: Demonstrates the registration steps for individual sensors, including editing configuration files and registration through a Web browser.
3. *Search for and listing of registered services and sensors*: Demonstrates the search and display functionalities for locating and listing services and individual sensors.
4. *Visualization of sensors locations on maps*: Demonstrates the map localization of registered sensors.
5. *Visualization of historical sensor data on charts*: Demonstrates the use of charts for visualizing historical observation data from individual sensors. The user can zoom in the chart and select a duration.
6. *Visualization of sensor event data on charts*: Demonstrates real-time visualization of events from individual sensors on charts.
7. *Composition of services*: Demonstrates the BPMN-based composition of registered SOS, WFS and WPS services.
8. *Data mediation*: Demonstrates how to specify data flow mapping for the composed SOS, WFS and WPS.
9. *Generation of WSDL and BPEL files for composed services:* Demonstrates the generation of WSDL and BPEL files for the composed BPMN model in the composition process.
10. *Publishing the composed model as a new resource*: Demonstrates the registration of the added-value service as a new resource in the platform, which can then be further used in compositions or for end-user applications.

3 Related Work, Summary, and Outlook

The huge amount of data generated by the increasing number of available sensor devices requires proper management in terms of abstraction, selection, and presentation in order to enable better decision making based on real-time sensor data.

Furthermore, development of added-value services based on such data needs to be faster and smarter. Sensapp aims to address these challenges by providing a sensor data/service management platform that combines open standards for abstracting interfaces from proprietary data and protocols, semantic technologies for better search and discovery, visual composition of services, and different data visualization techniques.

A working prototype of Sensapp with the functionalities presented in the demonstration section has been developed and is currently under performance evaluation. Some of the components such as the annotation, discovery, and execution components do not come with a graphical interface yet, but these are planned to be developed, possible with a close collaboration with the ENVISION project.[3] The source code of Sensapp is planned to be released as open source in the near future. As part of future work, the platform is planned to be deployed on the cloud and made available as a service for the wider community.

In enabling better access to real-time sensor data, Sensapp shares some of the ambitious of other initiatives such as HP Central Nervous System for the Earth (CeNSE) [9], Geospatial Cyberinfrastructure for Environmental Sensing (GeoCENS) [10], Nimbits [11], Pachube [12], Service Buss [13], or Hourglass [14]. Sensapp's focus on open standards as well as both on service developers and end-users, makes it a sensor integration platform that goes beyond the functionalities and scope of some of these approaches. Nevertheless, a detailed comparison with these existing approaches and possible synergies are part of future work.

Acknowledgments. This work is partly funded by the EU projects ENVISION (FP7-249120), ENVIROFI, REMICS, and the Norwegian national project Semicolon II.

References

1. OGC Sensor Observation Service (SOS), http://www.opengeospatial.org/standards/sos.
2. OGC Web Feature Service (WFS), http://www.opengeospatial.org/standards/wfs.
3. OGC Web Processing Service (WPS), http://www.opengeospatial.org/standards/wps.
4. OGC Sensor Event Service (SES), Position paper available at http://portal.opengeospatial.org/files/?artifact_id=29576.
5. W3C Resource Description Framework (RDF), Available at http://www.w3.org/RDF/.
6. OMG Business Process Modeling Notation (BPMN), Available at http://www.omg.org/spec/BPMN/2.0/.
7. OASIS Web Services Business Process Execution Language Version 2.0, Available at http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html.
8. JSR 286: Portlet Specification 2.0, http://www.jcp.org/en/jsr/detail?id=286.
9. HP Central Nervous System for the Earth (CeNSE), Information available via http://www.hpl.hp.com/news/2009/oct-dec/cense.html.
10. Geospatial Cyberinfrastructure for Environmental Sensing (GeoCENS), Information available via http://www.geocens.ca/.
11. Nimbits, Information available via http://www.nimbits.com/.
12. Pachube, Information available via https://pachube.com/.
13. The Sensor Bus, 52 North, Information available via http://52north.org/communities/sensorweb/incubation/sensorBus/.

[3] http://www.envision-project.eu/

Demonstration: Defining and Detecting Complex Events in Sensor Networks

Lucas Leidinger and Kerry Taylor

CSIRO ICT Centre
GPO Box 664, Canberra, ACT, 2602, Australia
kerry.taylor@csiro.au
lucas.leidinger@gmx.de

Abstract. Multi-modal sensor networks are difficult to program and difficult to use dynamically. We show how to use an ontology in the user interface to support end users to describe events of interest arising dynamically in sensor networks, to generate program code for the network devices to collect the necessary data, and to generate alerts when those described events are detected. The ontology is used for semantic optimisation at various points in the processing architecture.

Key words: Complex Event Processing, Ontology, Sensor Networks

1 Introduction

Improvements in technology and manufacturing reduce prices for sensing technology and allow the internet-connection of more and more base stations; therefore growing both the scale and application areas for wireless sensor networks. Reconfigurable, general purpose networks are replacing classic black-box sensor technology in the fields of environmental observations, for example conservation and disaster prevention, production, logistics, medicine and security.

Such networks provide high volumes of continuously captured measurement data. However, often only a small amount is interesting or has relevance for domain applications. Certain information may be required by different client applications, at several locations at the same time. The installation and configuration of sensor networks can quickly become complex, especially, if platforms and sensors of various manufacturers with different setups have to form a common network.

This demonstration deals with the idea of using semantic technologies and Complex Event Processing (CEP) to define and to detect complex events arising in the data collected by heterogeneous wireless sensor networks. A complex event has to be understood as a combination of filtered measurement values from particular sensors and locations in a well defined order within a specific period of time. The problem of programming the sensors and configuring the CEP system in several different low-level programming languages will be abstracted. The use of semantic ontologies allows the definition and detection of complex events independently of the type of sensor or kind of CEP system. For this purpose, event definitions, sensor programs and CEP queries are modelled in a custom ontology. The use of dynamic ontology assertions allows the recognition and reuse of existing sensor programs and available data streams. It makes it possible to perform semantic optimizations and to dynamically build a user control interface.

2 Architecture

To prove the idea of *ontology-driven complex event processing in heterogeneous sensor networks* and for demonstrating the use of semantic technologies combined with sensor networks, an *Event Framework* was developed.

An OWL2 *EventOntology* is the central part of the entire *Event Framework*. It is designed to store definitions of complex events and allows the use of reasoning and classification over event information to obtain additional knowledge and to perform semantic optimizations. Semantic optimizations for both sensor node programs and data streams are applied: ontology definitions of existing sensor programs and CEP streams are checked (by concept subsumption in the first case and concept membership in the latter) prior to creating new ones. If suitable pre-existing concepts are found in the ontology then the corresponding pre-existing sensor and instrument resources and CEP stream configurations, respectively, can be reused: saving instrument resources and reducing the amount of data which must be transferred between data source and event processing application. Hermit is used for the reasoning services.

The *User Interface*, an extension of the popular OWL ontology editor *Protégé*, allows one to define events in a logical and expressive way and to store this definition in an ontology. The entire complex event definition is composed of different parts: *Events*, *Alerts*, *Observations*, *Triggers* and *Sensor Programs* (see figure 1).

Semantic Event Middleware is the counterpart to the user interface. All created complex event descriptions which have been transformed into ontology data are forwarded from the *User Interface* to the *Semantic Event Middleware*. Here, the reverse process to transforming a user description into ontology data is performed.

To abstract the specification and access of event data streams, a *Management Module Interface* has been designed which allows the implementation of an independent solution for each specific kind of instrument or event source. Each distinct sensor network technology requires an implementation of the interface as a wrapper over the heterogenous aspects of the sensor network that are not modelled in the ontology (such as the native programming language grammar and the compilation/downloading tools).

The (*Coral8*[1] CEP-platform performs the actual complex event detection. For this, the server receives a stream with information about the event data

[1] The Coral8 CEP-platform. `http://www.aleri.com/products/aleri-cep/coral8-engine`

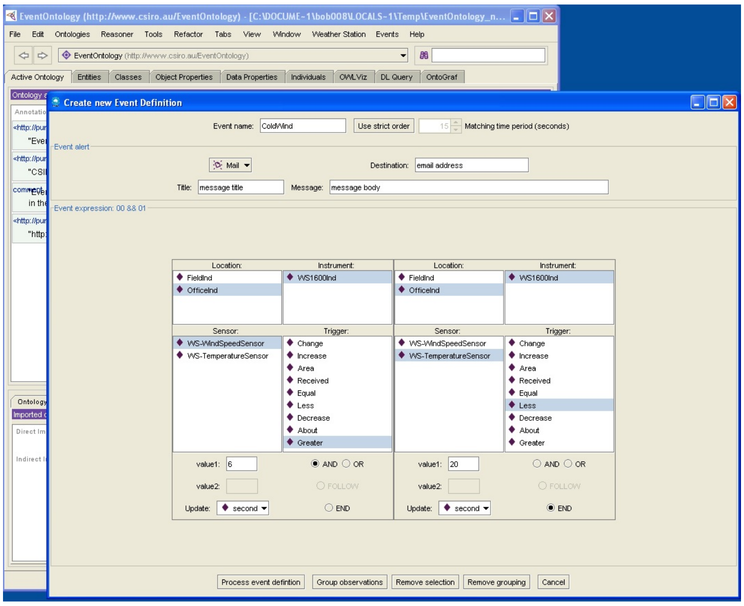


Fig. 1. User Interface

sources and a query which contains the program for the complex event detection. The CEP server is also responsible for sending the user defined alert message if an event has been detected.

3 Demonstration

The demonstration will show the interaction of the individual components of the *Event Framework*. While some sensor data will be simulated, we plan to also use a live Environdata WeatherMaster1600 instrument to make environmental observations and to show the *Event Framework*:

1. Showing the general functionality of the *User Interface*.
2. Defining a complex event definition composed from several atomic sensed events within the *User Interface*.
3. Processing the complex event definition: transformation into ontology data, programming selected instruments and start of the CEP event detection.
4. Simulating sensor data and changes in the environmental observation by the WeatherMaster1600 instrument.
5. Showing the detection of the defined event and the consequent alert.

4 Conclusion

The sensor programming function and it's ontology modelling allows high-level programming for sensor instruments, and can be used quite independently of the event detection function[3]. The event processing capability is described in more detail here[2].

The *Event Framework* is now being upgraded to work with an alternative data stream management system (GSN[1]) and will be deployed in February 2012, over a network of soil moisture sensors and smart ear-tagged cattle on a demonstration farm near Armidale, New South Wales, Australia. In this case it will be used as part of a system for precision agriculture: to alert the farmer to issues associated with livestock management (e.g. herd location, herd state), pasture management (e.g. plant water availability, pasture yield estimates), and joint management (e.g. that a herd should be moved from one paddock to another).

References

1. Karl Aberer, Manfred Hauswirth, and Ali Salehi. A middleware for fast and flexible sensor network deployment. In *Proceedings of the 32nd international conference on Very Large Data Bases*, VLDB '06, pages 1199–1202. VLDB Endowment, 2006.
2. Kerry Taylor and Lucas Leidinger. Ontology-driven complex event processing in heterogeneous sensor networks. In *The Semantic Web: Research and Applications: Proc. 8th Extended Semantic Web Conference*, volume 2 of *LNCS 6644*, pages 285–299, Heraklion, Greece, 29th May to 2nd June 2nd 2011. Springer.
3. Kerry Taylor and Patrick Penkala. Using explicit semantic representations for user programming of sensor devices. In *Advances in Ontologies: Proceedings of the Australasian Ontology Workshop*, volume 112 of *Conferences in Research and Practice in Information Technology*, Melbourne, Australia, 1 December 2009. Australasian Computer Society.

Demonstration: SECURE -- Semantics Empowered resCUe Environment

Pratikkumar Desai, Cory Henson, Pramod Anatharam, Amit Sheth

Kno.e.sis – Ohio Center of Excellence in Knowledge-enabled Computing,
Department of Computer Science and Engineering, Wright State University,
Dayton, OH 45435, USA
{pratik, cory, pramod, amit}@knoesis.org

Abstract. This paper demonstrates a Semantic Web enabled system for collecting and processing sensor data within a rescue environment. The real-time system collects heterogeneous raw sensor data from rescue robots through a wireless sensor network. The raw sensor data is converted to RDF using the Semantic Sensor Network (SSN) ontology and further processed to generate abstractions used for event detection in emergency scenarios.

Keywords: Semantic Sensor Web, abstraction, sensor, robotics, rescue environment, wireless sensor network

1 INTRODUCTION

Robots equipped with multiple heterogeneous sensors are quickly becoming an invaluable resource in emergencies and disaster scenarios [1]. They enable monitoring of the environment without unnecessarily risking the lives of first-responders. However, the avalanche of low-level sensor data generated by these robot sensors can quickly overwhelm the operator or decision-maker who is attempting to assess the situation. In this scenario, semantics can play a key role in interpreting the low-level sensor data and provide effective abstractions as a more intuitive representation of the situation; thus enabling the operator to make timely and effective decisions. In this paper, we describe a demonstration system that has a robot equipped with a temperature sensor, infrared sensor, and various gas sensors, such as carbon dioxide, carbon monoxide, methane, and compressed natural gas. With the help of Semantic Web technologies and domain-specific background knowledge, various different types of fires are detected, which will lead to different appropriate responses.

2 SYSTEM ARCHITECTURE

The purpose of SECURE is to detect fires of different classes –Class A (Ordinary combustibles), Class B (Flammable liquids), and Class C (Flammable Gases), etc. – within a building, utilizing sensor equipped robots. The detection of different types of fire can help the rescue workers to decide upon a proper response to the disaster. For example, different types of fires are extinguished differently depending on the composition of the combustible material.

Towards this goal, the SECURE system architecture is divided into four phases (as shown in Figure 1): (1) sensor data collection, (2) conversion of sensor observations to RDF, (3) analysis of sensor observations to generate abstractions, and (4) access through a graphical user interface.

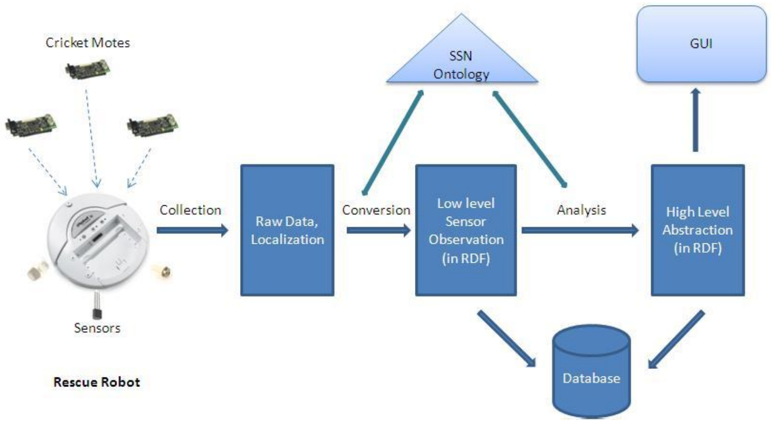


Figure 1. System Architecture

2.1 Sensor Data Collection

The sensor system consists of a wireless sensor network using Cricket motes [2] and a rescue robot equipped with environmental sensors. The cricket motes are deployed within the building, and use a combination of Time Difference of Arrival (TDoA) and multilitration for indoor location estimation [3]. The rescue robot is equipped with temperature sensors, infrared sensors, various gas sensors, and a cricket mote to provide location information.

2.2 Conversion of Sensor Observation to RDF

Data from the environmental sensors and indoor location information obtained from the cricket motes is continuously collected and streamed to a processing server. The raw sensor data is then converted to Resource Description Framework (RDF) [4] format using the OWLAPI[1] and Semantic Sensor Network (SSN) ontology [5].

2.3 Analysis of Sensor Observations to Generate Abstractions

Utilizing background knowledge which relates observable phenomena to different types of fires (encoded in a domain-specific ontology), the sensor observations are analyzed to determine the occurring fire event [6]. In this case, an abstraction is a record of the type of fire event; these abstractions are also encoded in RDF. The

[1] http://owlapi.sourceforge.net/

processed high-level abstractions will be used for situation awareness and decision making.

2.4 System Interface

The finalized results will be demonstrated using a simple GUI that will show to an operator the type of fire detected within a simulated environment. In addition, all the RDF data is published as Linked Data [7] and accessible through a SPARQL [8] endpoint.

3 DEMONSTRATION DESCRIPTION

As described in the Figure 2, the potential scenario includes a building structure with various chemicals stored in different rooms. Due to an accident, the building catches fire while the source of the fire is unknown to the first responders. The workshop demonstration will involve a simplified version of the scenario shown in Figure 2, with live-video of a robot carrying variety of sensors described above, and sensing a fire within a building, and in real-time, updating a GUI with the sensor readings and event detection, specifically the type of fire. Due to the challenge of creating various types of fire inside building, the simplified goal of the demonstration will be to differentiate sources of heat such as a candle, a butane stove or portable heater with the location information.

4 CONCLUSION

SECURE provides a layered approach for event detection in emergency scenarios to avoid information overload and improved decision-making for the emergency response operators. Figure 2 illustrates an example scenario where various types of chemicals are stored in a building, and there is a fire of unknown type. To deal with such a situation, the system consists of a rescue robot equipped with heterogeneous sensors and assisted with a wireless sensor network based indoor location system. The raw sensor data is processed using SSN ontology and background knowledge to convert the raw data to actionable abstractions representing a more intuitive representation of the situation.

REFERENCE

[1] Albert K., Henry Y.K., Robot Assisted Emergency search and Rescue System with a Wireless Sensor Network, International Journal of Advanced Science and Technolgy, 2009.
[2] Nissanka B. Priyantha, Anit Chakraborty, Hari Balakrishnan,The Cricket Location Support System, Proc. 6th ACM MOBICOM, Boston, MA, August 2000.
[3] P. Desai., Localization and Surveillance using Wireless Sensor Network and Pan/Tilt Camera, M.S. Thesis, Department of Electrical Engineering,Wright State University, 2009.

[4] Manola, F., and Miller, E. (2004). RDF Primer. (Frank Manola & Eric Miller, Eds.) W3C Recommendation (Vol. 10, pp. 1-107). W3C.
[5] Lefort, L., Henson, C., Taylor, K., Barnaghi, P., Compton, M., Corcho, O., Garcia-Castro, R., Graybeal, J., Herzog, A., Janowicz, K., Neuhaus, H., Nikolov, A., and Page, K.: Semantic Sensor Network XG Final Report, W3C Incubator Group Report (2011). Available at http://www.w3.org/2005/Incubator/ssn/XGR-ssn/
[6] Henson, C., Thirunarayan, K., Sheth, A., and Hitzler, P. Representation of Parsimonious Covering Theory in OWL-DL. In: Proceedings of the 8th International Workshop on OWL: Experiences and Directions (OWLED 2011), San Francisco, CA, United States, June 5-6, 2011.
[7] Bizer, C., Heath, T., and Berners-Lee, T.. Linked Data - The Story So Far. International Journal on Semantic Web and Information Systems, 5(3), 1-22. Elsevier. 2009. doi: 10.4018/jswis.2009081901.
[8] Prud'hommeaux, E., and Seaborne, A. (2008). SPARQL Query Language for RDF. (E. Prud'hommeaux and A. Seaborne, Eds.) W3C Recommendation. W3C.

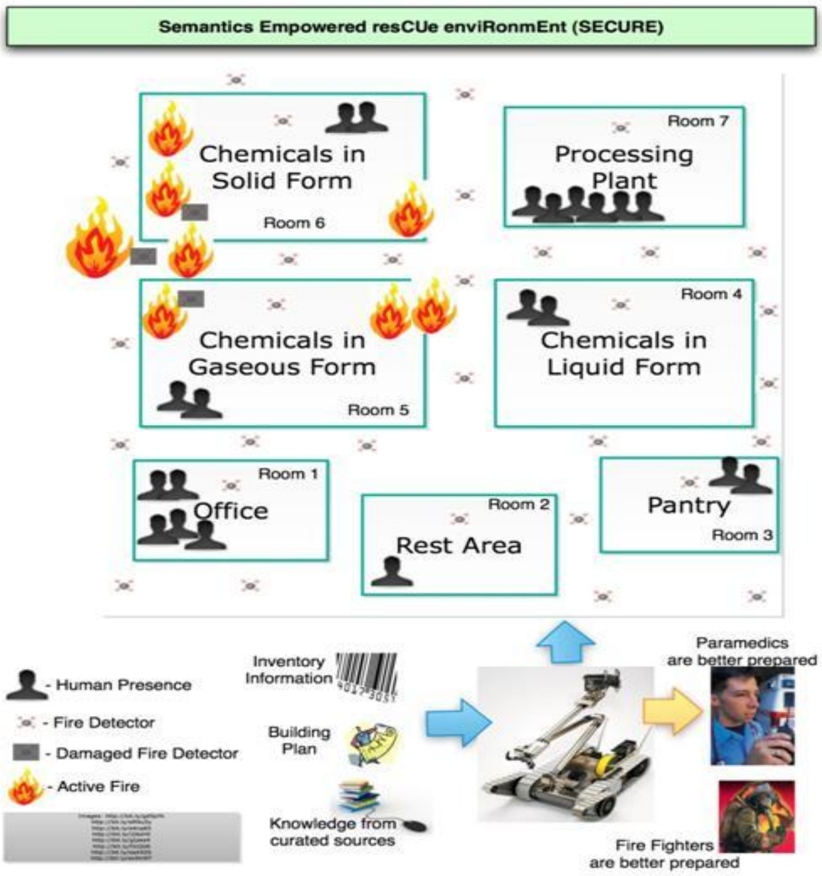


Figure 2. Example emergency scenario

Demonstration: Real-Time Semantic Analysis of Sensor Streams

Harshal Patni, Cory Henson, Michael Cooney, Amit Sheth, Krishnaprasad Thirunarayan

Kno.e.sis – Ohio Center of Excellence in Knowledge-enabled Computing
Department of Computer Science and Engineering, Wright State University
Dayton, OH 45435, USA
{harshal, cory, michael, amit, tkprasad}@knoesis.org

Abstract. The emergence of dynamic information sources – including sensor networks – has led to large streams of real-time data on the Web. Research studies suggest, these dynamic networks have created more data in the last three years than in the entire history of civilization, and this trend will only increase in the coming years [1]. With this coming data explosion, real-time analytics software must either adapt or die [2]. This paper focuses on the task of integrating and analyzing multiple heterogeneous streams of sensor data with the goal of creating meaningful abstractions, or features. These features are then temporally aggregated into feature streams. We will demonstrate an implemented framework, based on Semantic Web technologies, that creates feature-streams from sensor streams in real-time, and publishes these streams as Linked Data. The generation of feature streams can be accomplished in reasonable time and results in massive data reduction.

Keywords: Streaming Sensor Data, Abstraction, Semantic Web, Semantic Sensor Web

1 Introduction

Sensors produce huge amounts of low-level data about our environment that arrives in the form of rapid, continuous, and time-varying streams [3]. These data streams could quickly overwhelm any system not capable of effectively detecting and analyzing the most important data. Analyzing such sensor data streams and providing meaningful abstractions in real-time presents a significant research challenge. An abstraction, also called a feature, is a high-level representation of low-level sensor data.

There has been a lot of work in the database community on analyzing and mining real-time streaming data. Most of the current approaches within the database community provide mathematical summaries (i.e., minimum, maximum, average and count) for a single modality stream (like a temperature stream) over time (i.e. within a time window) [4]. These summaries are necessary and useful, but provide little help in answering questions involving real world events, such as: *Which weather stations are currently detecting a Blizzard?* Or: *What event (or sequence of events) is currently being detected by a weather station?*

The ability to answer such questions requires the semantic integration and inference over data from multiple single modality sensor streams using external domain knowledge. A feature-stream can be generated by aggregating a sequence of features detected by a particular sensor (or set of sensors), over a period of time. Feature-streams provide a clear and intuitive representation of how events evolve over time. An intuitive representation of trends in features will present decision makers with actionable situation awareness.

2 System Architecture

Consider the following question: *What weather events are currently being detected near Dayton James Cox Airport?* In order to answer this question, we would first need to find sensors near Dayton James Cox Airport, then access data streams for these sensors, integrate the streams capable of detecting the weather events, and finally, detect and represent the events.

The generation of feature streams requires a framework that can generate, integrate, and reason over multiple heterogeneous sensor streams. Reasoning over the integrated streams uses background knowledge and rules to generate feature-streams that represent events in the real world. The feature-stream framework is divided into four parts (see figure 1): (1) raw data generation, (2) data stream generation, (3) feature-stream generation, and (4) feature stream access.

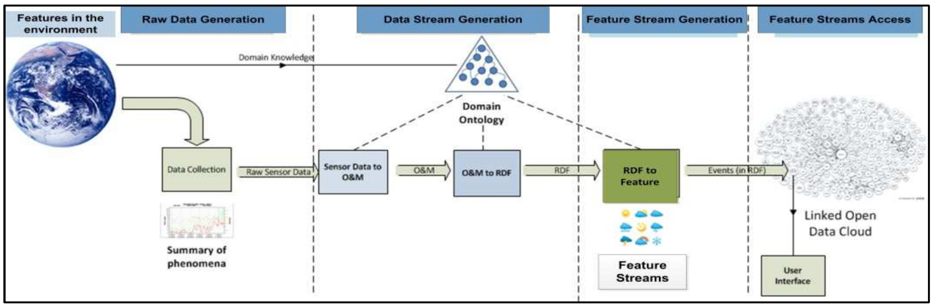


Fig. 1. Framework for generating Feature Streams

1 **Raw Data Generation:** The framework begins with the collection of raw streaming data from sensors within an environment. In this demonstration, we utilize MesoWest[1] , a project within the Department of Meteorology at the University of Utah, which provides near real-time access to weather sensor streams using a service API. Observations provided by MesoWest are encoded as CSV text, and includes measurements for temperature, visibility, precipitation, pressure, wind speed, humidity, etc. Example data provided by MesoWest can be seen below. The example contains information regarding the date and time of the observation, along with temperature (TMPF), wind speed (SKNT), and precipitation (PREC) observation values

```
PARMETER = MON, DAY, YEAR, HR, MIN, TMZN, TMPF, SKNT, PREC
VALUE = 11, 5, 2010, 13, 50, PDT, 30, 37, snow
```

2 **Data Stream Generation:** The second phase converts the stream of raw sensor data into an RDF stream. The raw sensor stream obtained from MesoWest is initially converted to Observation and Measurements (O&M)[2] format. O&M is a well-accepted XML standard in the sensors community. The SAX (Simple API for XML) parser[3] is used to generate the O&M XML stream. Below is an example encoding of the temperature, wind speed, and precipitation observations in O&M. The observation values for different time instants are separated using a block separator @@.

```
<swe:encoding>
```

[1] http://mesowest.utah.edu/
[2] http://www.opengeospatial.org/standards/om
[3] http://www.saxproject.org/

```
   <swe:TextBlock decimalSeparator="." tokenSeparator="," blockSeparator="@@"/>
</swe:encoding>
<swe:values>2010-5-11T13:50:00,30,37,snow@@</swe:values>
```

The O&M stream is then converted to an RDF[4] stream. RDF is a Semantic Web standard model for representation and interchange of data on the Web. XSLT[5] is used to convert the O&M to RDF, conformant to the W3C Semantic Sensor Network (SSN) ontology [6]. Below is an example RDF encoding of a temperature observation. [Note that *ssn*, *weather*, and *time* correspond to the prefixes for the SSN ontology, weather ontology, and OWL-Time ontology, respectively; *ssn-weather* corresponds to individuals generated by the system.]

```
ssn-weather:Observation_Temperature_KDAY_2005_10_21_5_30
   a ssn:Observation ;
   ssn:observedProperty weather:Temperature ;
   ssn:observedBy ssn-weather:System_KDAY ;
   ssn:observationResult ssn-weather:MeasureData_Temperature_KDAY_2010_05_11_13_50 ;
   ssn:observationSamplingTime ssn-weather:Instant_2010_05_11_13_50 .

ssn-weather:MeasureData_Temperature_KDAY_2010_05_11_13_50
   a ssn:SensorOutput ;
   ssn:hasValue "30.0" ;
   weather:uom weather:fahrenheit .

ssn-weather:Instant_2010_05_11_13_50_00
   a time:Instant ;
   time:inXSDDateTime "2010-05-11T13:50:00" .
```

3 **Feature Stream Generation:** The third phase integrates the RDF sensor streams and reasons over the integrated streams to detect features. Feature definitions are obtained from National Oceanic and Atmospheric Administration (NOAA)[6], and defined in the weather ontology. The feature definitions are initially used to filter the sensors capable of detecting a feature. A sensor is capable of detecting a feature if it is capable of observing all the phenomena that compose a feature. Filtering improves performance by reducing the number of sensor streams that are reasoned upon. SPARQL[7] is used for reasoning over the integrated sensor streams. An example SPARQL rule for detecting a Flurry over weather station KDAY is given below.

```
PREFIX ssn-weather:<http://knoesis.wright.edu/ssw/ont/ssn-weather.owl1#>
PREFIX ssn:<http://http://purl.oclc.org/NET/ssnx/ssn/>
PREFIX weather:<http://knoesis.wright.edu/ssw/ont/weather.owl#>

ASK
{
   ?windSpeedObs ssn:observedBy ssn-weather:System_KDAY .
   ?windSpeedObs ssn:observedProperty weather:WindSpeed .
   ?windSpeedObs ssn:observationResult ?windSpeedResult .
   ?windSpeedResult ssn:hasValue ?windSpeedValue .

   ?snowObs ssn:observedBy ssn-weather:System_SB1 .
   ?snowObs ssn:observedProperty weather:Snowfall .
   ?snowObs ssn:observationResult ?snowResult .
   ?snowResult ssn:hasValue ?snowValue .

   FILTER(?windSpeedValue < 35)
   FILTER(?snowValue = "true")
}
```

The SPARQL rule is used to detect the most recent/current feature. A sequence of features detected over time results in a feature stream.

[4] http://www.w3.org/RDF/
[5] http://www.w3.org/TR/xslt
[6] http://www.noaa.gov/
[7] http://www.w3.org/TR/rdf-sparql-query/

4 **Feature Stream Access:** Finally, the feature stream is published as Linked Data [5]. The features can be accessed using either directly by issuing SPARQL queries to the RDF or through a map-based GUI[8].

3 Demonstration

During the workshop, a Google Maps based GUI will be demonstrated, showcasing the generated feature streams. The user can either select all the weather stations in a state, or search for a station by named location (using Geonames). Next the user is provided with an option to select features of interest. The system can currently detect blizzard, flurry, rain shower, and rain storm. Feature selection will result in the filtering of stations that are able to detect the features of interest. Clicking on a station shows the features detected over time along with the associated lower-level sensor observations. Because the features may not occur in real-time at the demonstration time, we will have a backup providing examples of interesting past events.

4 Evaluation

To evaluate the performance of this system, we collected 120 hours of data for sensors in (and around) Utah between February 2^{nd} to 6^{th} 2003. . Figure 2 shows an average of the amount of time (in ms) taken for each phase during feature generation. On average, for each hour, 427 sensors provided data during the evaluation, and produced an average of 1104 observations. 9 flurries, 1 rain shower, and 417 clear features were detected during the evaluation. We found an order of magnitude distinction between the number of observations and feature generated, which means storing only the features (if applicable) would result in massive data reduction. A demonstration page[9] will provide more details, including the storage evaluation

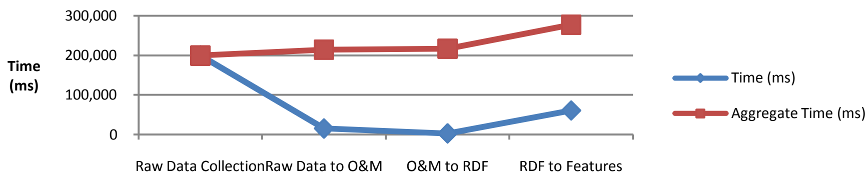


Fig. 2. Performance Evaluation over Time

References

[1]. Gigaom Aricle on Big Data, 2010, http://gigaom.com/cloud/sensor-networks-top-social-networks-for-big-data-2/

[2]. Software must adapt or Die, 2010, http://www.readwriteweb.com/archives/data_analytics_software_must_adapt.php

[3]. Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom *Models and Issues in Data Stream Systems,* In Proceedings of the 21st ACM Symposium on Principles of Database Systems, 2002.

[4]. Refik Samet and Serhat Tural. 2010. *Web based real-time meteorological data analysis and mapping information system*. In Proceedings of *WSEAS Transactions of Information Science. and Applications,* September 2010, 1115-1125.

[5]. Bizer, C., Heath, T., and Berners-Lee, T. *Linked Data – The Story* So Far. International Journal on Semantic Web and Information Systems, 5(3), 1-22. Elsevier. 2009.

[6]. Lefort, L., Henson, C., Taylor, K., Barnaghi, P., Compton, M., Corcho, O., Garcia-Castro, R., Graybeal, J., Herzog, A., Janowicz, K., Neuhaus, H., Nikolov, A., and Page, K.: Semantic Sensor Network XG Final Report, W3C Incubator Group Report (2011). Available at http://www.w3.org/2005/Incubator/ssn/XGR-ssn/

[8] http://knoesis1.wright.edu/EventStreams

[9] http://wiki.knoesis.org/index.php/SSN_Demo

Demonstration: A RESTful SOS Proxy for Linked Sensor Data*

Arne Bröring[1], Krzysztof Janowicz[2], Christoph Stasch[3], Sven Schade[4], Thomas Everding[3], and Alejandro Llaves[3]

[1] 52°North Initiative for Geospatial Open Source Software GmbH, Germany
[2] University of California, Santa Barbara, USA
[3] Institute for Geoinformatics, University of Münster, Germany
[4] Institute for Environment and Sustainability, Joint Research Centre, Ispra, Italy

Abstract. Next generations of spatial information infrastructures call for more dynamic service composition, more sources of information, as well as stronger capabilities for their integration. Sensor networks have been identified as a major data provider for such infrastructures, while Semantic Web technologies have demonstrated their integration capabilities. Most sensor data is stored and accessed using the Observations & Measurements (O&M) standard of the Open Geospatial Consortium (OGC) as data model. However, with the advent of the Semantic Sensor Web, work on an ontological model gained importance within Sensor Web Enablement (SWE). The ongoing paradigm shift to Linked Sensor Data complements this attempt and also adds interlinking as a new challenge. In this demonstration paper, we briefly present a Linked Data model and a RESTful proxy for OGC's Sensor Observation Service (SOS) to improve integration and inter-linkage of observation data.

Keywords: Semantic Sensor Web, Linked Sensor Data, REST, Sensor Observation Service

1 Introduction

The Sensor Web requires well defined semantics to make observation data discoverable and reusable [2]. The Semantic Web provides the necessary framework by (i) formal and machine-readable ontologies for sensors, observations, and observed properties, and by (ii) using reasoning to discover implicit facts, relations, and contradictions. So far, the Sensor Web and Semantic Web are not well connected which limits data exchange as well as combining their services. To address this problem, we have proposed and partially implemented a Semantic Enablement Layer for Spatial Data Infrastructures (SDI) [3]. It encapsulates Semantic Web reasoners and ontology repositories within OGC Web services to enable a transparent and seamless integration of Semantic Web technologies with SDIs. This work focuses on enabling the reverse direction, i.e., making spatial information available on the Semantic Web without changing existing standards and implementations. To facilitate integration and inter-linkage of observation data, this

* This demonstration paper is a modified excerpt of the article by *Janowicz et al. 2011* [1]

demonstration paper presents a Linked Data model and a RESTful proxy for the Sensor Observation Service (SOS) interface of OGC's Sensor Web Enablement initiative [4]. For two related approaches on serving semantic-enabled sensor data see [7,8].

2 System Architecture

The RESTful SOS proxy is available as free and open source software[5]. It can be installed as a software facade in front of any OGC conform SOS and offers the core functionality to make sensor data available as Linked Data. Based on a well-defined URI scheme [1], the RESTful proxy extracts the user's query from the URI, encodes it into valid SOS queries, fetches the results from the underlying SOS, and converts them (after content negotiation) to RDF/XML aligned with the developed model for Linked Sensor Data (Figure 2). Consequently, each URI identifies a particular data set and at the same time encodes a query to the underlying SOS.

The RESTful SOS proxy is implemented using the *OX-Framework* [5], a software framework which facilitates the utilization of OGC Web Services, such as the SOS. The OX-Framework handles access of various service interfaces by providing a generic architecture that includes a plug-in mechanism for service adapters as extension points of the framework.

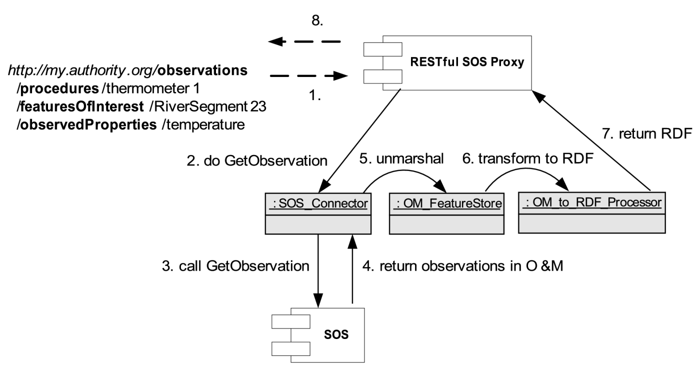


Fig. 1. Resolving a URI by the RESTful SOS proxy [1]

Three kinds of service adapters are needed for accessing a service (Figure 1): *Service connectors* trigger service operations and instantiate the common capabilities model. *Feature stores* provide the functionality to unmarshal received feature data into the internal feature model of the OX-Framework, while *data processors* run on the instantiated feature model and transform the feature data into other representations. We developed a data processor that converts observations into RDF-encoded Linked Data; however, we also support other representations such as KML or JPEG charts. The

[5] http://52north.org/RESTful_SOS

RESTful SOS proxy chooses the right data processor based on HTTP content negotiation.

3 Demonstration

The proxy exposes sensor data following a particular URI scheme. While OGC's Observations & Measurements standard supports unique identifiers, it currently does neither prescribe the use of HTTP URI's, the persistence of identifiers, nor clear and flexible linking strategies between resources. Ontologies abstract from data models and aim at describing the physical world. For example, they specify the notion of a stimulus which triggers a sensor and leads to the observation. The stimulus as such, however, is out of scope for O&M. Therefore, we introduce an intermediate Linked Data model by extending the W3C SSN ontology's Stimulus-Sensor-Observation (SSO) ontology design pattern [6]; see Figure 2. The relations between the presented classes act as links in our model and define the multiple navigation paths and external references.

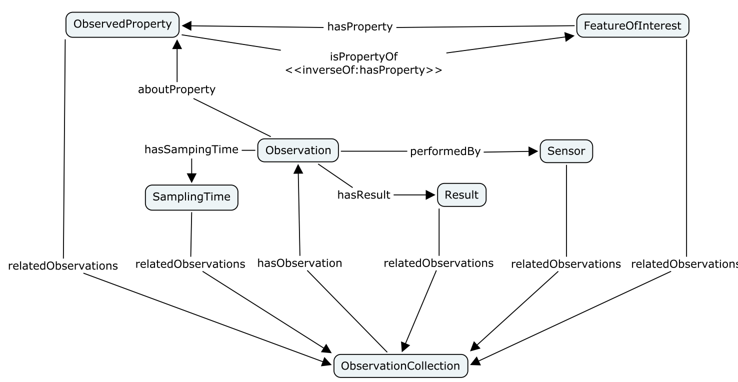


Fig. 2. Concept map with the classes and relations of the Linked Sensor Data model [1].

In the demonstration, we present how URIs act as identifiers for sensor data and as query filters which are mapped by the RESTful proxy to SOS GetObservation requests. For instance, the URI `http://v-swe.uni-muenster.de:8080/52nRESTfulSOS/RESTful/sos/AirBase_SOS/observations/sensors/HR:0002A/samplingtimes/2008-01-01,2008-12-31/observedproperties/concentration[NO2]` points to the observation collection with all NO_2 observations from a specific sensor during 2008.As the proposed solution offers the sensor data as a RESTful service, we will apply a common web browser to illustrate how queries are constructed and how users may interact with the service front-end.

4 Conclusion

In this demonstration paper, we report on the implementation of a transparent and RESTful SOS proxy that can serve Linked Sensor Data without any modifications to existing OGC services and existing SDI deployments. We decided to use a RESTful approach as it combines three key advantages. First, URIs are building blocks of Linked Data. REST allows us to identify data and at the same time encode the query using our URI scheme. Second, a major requirement of our vision of Semantic Enablement [3] is transparency, which is given by our REST proxy approach. Third, the REST paradigm focuses on simplicity with respect to application implementation.

Summing up, the proposed approach provides an important step towards the semantic enablement of existing information systems and infrastructures, and thereby eases the integration of dynamic information sources such as sensor networks. Delivering observations as Linked Data, connecting them with other data sources, and using ontologies and Semantic Web reasoners to improve retrieval, alignment, and matching are major building blocks for the implementation of novel information infrastructures.

Acknowledgments

The presented work is developed within the 52° North semantics community (http://52north.org/semantics), and is partly funded by the European projects *EO2Heaven* (FP7-244100) and *ENVIROFI* (FP7-284898).

References

1. Janowicz, K., Bröring, A., Stasch, C., Schade, S., Everding, T., Llaves, A.: A restful proxy and data model for linked sensor data. International Journal of Digital Earth (2011; in press)
2. van Zyl, T., Simonis, I., McFerren, G.: The sensor web: systems of sensor systems. International Journal of Digital Earth **2 (1)** (2009) 16–30
3. Janowicz, K., Schade, S., Bröring, A., Keßler, C., Maue, P., Stasch, C.: Semantic enablement for spatial data infrastructures. Transactions in GIS **14**(2) (2010) 111–129
4. Bröring, A., Echterhoff, J., Jirka, S., Simonis, I., Everding, T., Stasch, C., Liang, S., Lemmens, R.: New Generation Sensor Web Enablement. Sensors **11**(3) (2011) 2652–2699
5. Page, K., De Roure, D., Martinez, K., Sadler, J., Kit, O.: Linked sensor data: Restfully serving rdf and gml. In K., T., Ayyagari, A., De Roure, D., eds.: Proceedings of the 2nd International Workshop on Semantic Sensor Networks (SSN09). Volume Vol-522., CEUR (2009) 49–63
6. Henson, C.A., Pschorr, J.K., Sheth, A.P., Thirunarayan, K.: Semsos: Semantic sensor observation service. In: International Symposium on Collaborative Technologies and Systems (CTS 2009), Baltimore, MD, USA, IEEE (May 2009)
7. Bröring, A., Jürrens, E.H., Jirka, S., Stasch, C.: Development of Sensor Web Applications with Open Source Software. In: First Open Source GIS UK Conference (OSGIS 2009), 22 June 2009, Nottingham, UK. (2009)
8. Janowicz, K., Compton, M.: The stimulus-sensor-observation ontology design pattern and its integration into the semantic sensor network ontology. In Ayyagari, A., Roure, D.D., Taylor, K., eds.: Proceedings of the 3rd International workshop on Semantic Sensor Networks 2010 (SSN10) in conjunction with the 9th International Semantic Web Conference (ISWC 2010). Volume 668., CEUR-WS (2010)