

A Semantic-based Architecture for Sensor Data Fusion

Anastasios Zafeiropoulos, Nikolaos Konstantinou, Stamatios Arkoulis,
Dimitrios-Emmanuel Spanos, Nikolas Mitrou
National Technical University of Athens,
9, Heroon Polytechniou Str., 15773 Zografou, Athens, Greece
{tzafeir, nkons, stark, dspanos}@cn.ntua.gr, mitrou@softlab.ntua.gr

Abstract

In this paper, we present a three-layer flexible architecture which intends to help developers and end users to take advantage of the full potential that modern sensor networks can offer. The proposed architecture deals with issues regarding data aggregation, data enrichment and finally, data management and querying using Semantic Web techniques. Semantics are used in order to extract meaningful information from the sensor's raw data and thus facilitate smart applications development over large-scale sensor networks. We describe an open, standards-based, modular architecture which is based on the current standardization efforts of the Open Geospatial Consortium (OGC) and extends them to support Semantic Web techniques comprising a core of reusable components and interfaces for supporting different types of services, through Web technologies.

1. Introduction

Sensor networks have attracted a lot of attention lately due to advances in the area of microsensors. A vast amount of small, inexpensive, energy-efficient, and reliable sensors with wireless networking capabilities is available worldwide increasing the number of the sensor network deployments [16]. Furthermore, the deployment of IPv6 provides a huge address space for networking purposes in order to address the large sensor networks on a global scale and leads to the rapid deployment of many useful applications. It is not unreasonable to expect that in the near future, many segments of the networking world will be covered with wireless sensor networks (WSNs) that will be accessible via the Internet.

Nevertheless, as stated in [4], too much attention has been placed on the networking of distributed sensing and too little on tools to manage, analyze, and understand the collected data. Taking under

consideration that the amount of data available through a huge number of connected heterogeneous sensors is increasing, an architecture is necessary for proper data management that will process the collected information in order to provide meaningful results and finally user-defined actions and alarms.

Furthermore, data management in sensor networks is very important due to the resource constraints imposed by the sensor nodes, such as low battery power, limited signal processing, limited computation and communication capabilities, and a small amount of memory. In order to succeed proper data management, aggregation and processing of data have to be done in a way that renders them helpful to applications that receive stored or real time input and undertake specific actions. The aggregated data have little meaning by themselves, therefore, in order for the event detection to be done, they have to be interpreted according to the information which is important for the final user. Furthermore, in many cases data from many heterogeneous data sources might need to be compared and/or combined and different events have to be correlated. Thus, in accordance with the type of sensors, suitable schemes for aggregation of data have to be deployed in order for the data to be processed in central nodes with increased capabilities.

Given the successful data aggregation, the technical challenges of extracting meaningful events and enabling interoperability among data from different sources can be addressed using the Semantic Web. The basic concept of the Semantic Web contribution in this aspect is content annotation. Data existing either on the Web or in restricted access repositories should be annotated in order to become retrievable. This purpose is served by a number of prevalent Semantic Web technologies like content description languages, query languages and annotation frameworks. Semantic annotation in the form of metadata can be added to any form of context, in order to add well-defined semantics that will ease its use. In other words, Semantic Web

can connect the sensory data to the features of their environment.

Taking into account the above considerations, throughout this paper we analyze and discuss the problem of information gathering, processing and intelligently exploiting real-time or recorded streams of data produced by mostly heterogeneous sources of sensors and we present a flexible and lightweight architecture which can be easily deployed for this purpose. The distinguishing characteristic of this architecture is that both low level components such as sensors, middle level components such as XML templates for information tagging, and high level components such as application-specific ontologies are pluggable to the architecture, thus facilitating application development and system configuration to different real-world scenarios.

2. Related Work – Motivation

2.1. Existing Systems and Applications

Many existing deployments and applications are available nowadays that are trying to exploit the recent advances in sensor networking. In the next paragraphs we present some proposed architectures that are engaged with data aggregation and processing in sensor networks, a number of which support Semantic Web technologies, and finally we declare the motivation for the proposition of our architecture.

One of the major efforts that is worth mentioning is the Sensor Web Enablement (SWE) initiative, initiated by the OGC¹. The goal of this initiative is the development of standards that will enable the discovery, exchange, and processing of sensor observations, as well as the tasking of sensor systems. However, for the time being, there is no explicit ontological structure proposed in the SWE framework. Although all services make use of a common encoding and transport protocol, semantic interoperability is still an issue due to the lack of an explicit common formal conceptual model [17].

Contrarily, the ES3N architecture [12] applies Semantic Web technologies on top of a sensor network. Its goal is to develop an ontology-based storage mechanism for sensor observations that will let the end user of the system to post semantic queries. This is accomplished through the use of an RDF repository containing daily records of all sensor measurements. Then, rudimentary SPARQL queries

are posed in order to extract a specific observation. However, the performance and efficiency of the system when the volume of data increases are questionable, given that the response time for the SPARQL queries can rise to unacceptable levels.

The IrisNet architecture [7] provides a software infrastructure that lets users query globally distributed collections of high bitrate sensors powerfully and efficiently. IrisNet envisions a worldwide sensor web, in which users can query, as a single unit, vast quantities of data from thousands or even millions of widely distributed, heterogeneous sources. This is achieved through a network of agents, responsible for the collection and storage organization of sensor measurements. Data are stored in XML format and XPath is used as the means to answer user queries. This choice, however, is highly restrictive, decreasing the reusability of IrisNet, which could largely benefit from the use of Semantic Web techniques.

In the same direction, the SWAP framework [14] is another effort to combine data from sensors in order to carry out high-level tasks and offer to the end-user a unified view of the underlying sensor network. SWAP proposes a three tier architecture comprising a sensor, a knowledge and a decision layer, each one of them consisting, in turn, of a number of agents. Special care is taken for the semantic description of the services available to the end-user, allowing the composition of new applications. Nevertheless, the use of multiple agents and ontologies which are related to the offered services of each agent increase substantially the complexity of such a system.

In our previous work, we have described Priamos [11] which is a middleware architecture for automated, real time, unsupervised annotation of low-level context features and their mapping to high-level semantics. A prototype implementation based on Semantic Web tools is presented in depth, while the benefits and drawbacks of this approach are underlined. We argue that the use of Semantic Web provides powerful answers to context awareness challenges. Furthermore, Priamos enables the composition of simple rules through specific interfaces, which may launch a context aware system that will annotate content without the need for user technical expertise.

2.2. Data Transformation and Semantic Representation

Due to the raw nature of sensory data and the fact that they cannot provide us with high-level information extraction, specific models are being used in order to interpret them. Each model is provided depending on the scope of the application that will use it and the kind

¹ The Open Geospatial Consortium (<http://www.opengeospatial.org>) is an international consortium of industry, academic and government organizations, working on the development of open standards for geospatial and location services.

of information that it requires. Different XML templates can interpret in a different way the sensors data according to the application related view.

In this direction, OGC members are specifying interoperability interfaces and metadata encodings that enable real time integration of heterogeneous sensor webs into the information infrastructure. OGC members have developed and tested among others the following candidate specifications:

- Observations & Measurements (O&M) – Standard models and XML Schema for encoding observations and measurements from a sensor, both archived and real-time.
- Sensor Model Language (SensorML) - Standard models and XML Schema for describing sensor systems and processes associated with sensor observations (discovery of sensors, location of sensor observations, processing of low-level sensor observations).

In our proposed architecture, we will follow these specifications without rendering them obligatory to everyone who wants to use other XML schemas.

Additionally, the semantic representation of sensor networks data is very significant because an ontology specifies the important concepts in a domain and how they are related, giving structure to the knowledge about a certain domain. In [6], a universal ontology for describing concepts and relationships of the sensor networks units and data has been proposed. The benefits of using this ontology are the maximization of the precision of searching sensor data by utilizing the semantic information. Furthermore, in [18], a practical approach to the development of comprehensive sensor ontologies based upon deep knowledge models rather than capturing only superficial sensor attributes is presented.

2.3. Motivation

As stated earlier, many approaches are available today for managing sensor networks, regarding especially the aggregation and processing of data. However, the downside of these approaches are the limitations regarding the size of the sensor network, the amount of data that can be transferred, the support of distributive sensor deployments and the lack of semantic data representation. It is questionable whether these approaches will scale with the mass increase in the deployment of heterogeneous sensor networks and the abilities for remote communication and management of the sensor resources.

In such an environment, the lack of context annotation with semantics and the absence of ontological infrastructures for high-level rules and

queries, restricts the potential of end users to exploit the acquired information, to match events from different sources and to deploy smart applications which will be capable of following semantic-oriented rules. Our proposed architecture tries to tackle with this issues and to provide a flexible and modular scheme for easily deploying and managing large scale sensor networks, consisting of heterogeneous data sources.

3. Proposed Architecture

In this section the proposed architecture is described analytically. The architecture contains three layers: the Data layer, the Processing layer and the Semantic layer, each of which is presented in detail in the following subsections (see Figure 1).

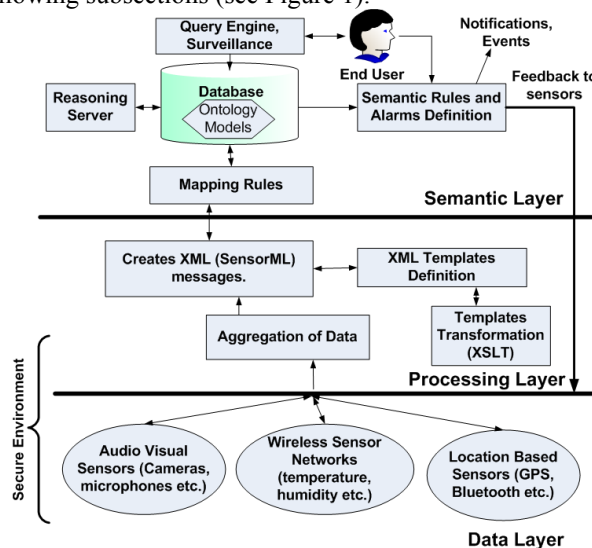


Figure1. Abstract Overall Architecture

3.1. Data Layer

This layer describes how the raw sensor data can be discovered, collected and aggregated in a central entity. The central entity will be responsible for the discovery of new nodes and the specification of the data acquisition policy. The data acquisition can be event-based where data are sent from the source and a method is called to collect them (serial ports, wireless cameras) or polling based where the central node periodically queries the data from the managed sensors. In order to be able to define the collection process, we have to classify the sensors in the following categories: Wireless Sensors, Location Sensors and Audiovisual Sensors.

The first category refers to wireless sensors measuring and monitoring environmental metrics (e.g. temperature, humidity, acceleration, pressure). Many proposed architectures, routing protocols and routing

schemes can be deployed for data aggregation, due to the need for low energy consumption and network congestion avoidance. In most of them, the data are either routed to specific nodes inside the network and then forwarded to the central infrastructure, or nodes send their data directly to a number of central entities. Regarding the way that these data should be transmitted to the next layers, it can be done by custom applications, based on TCP/IP protocol stack, which utilize technologies like Java/C Sockets, HTTP(S) Connections, Bluetooth Serial Drivers or Web-Services.

The second category refers to smart phones, PDAs and other handheld or embedded devices with installed positioning interfaces (Bluetooth, GPS, Wi-Fi). These devices collect location-sensitive data and send them, either directly or via special infrastructures, to the data aggregators. The aggregators may be reached by these devices using custom applications written in J2ME, Symbian C++, C# and .NET with WAP, GPRS, and Bluetooth or Wi-Fi as their communication channel.

The last category contains sensors like video cameras or microphones. Because of the nature of the collected data a huge amount of bits may be transmitted through wired or wireless infrastructures to the next layer of our architecture, and in many cases special networking requirements, such as required bandwidth, jitter and packet loss have to be satisfied. The data offered by such sensors become available through either Web Servers or Web Services technology.

In addition to raw data aggregation, security awareness is an important aspect of our proposed architecture due to the reason that sensor networks often store, process and interact with sensitive data in hostile and unattended environments.

Security in WSNs has attracted the interest of many researchers and, thus, the requirements (data confidentiality / integrity / freshness, availability, self-organization, time synchronization, secure localization, authentication) and the obstacles (inherent resource and computing constraints, unreliable communication, unattended operation) for security have been defined. The current attacks have been carefully classified and the corresponding defensive measures have already been listed in detail [1]. In cases where the location of important sensor-observed aspects must be hidden, anonymity mechanisms [10] have to be implemented in order to ensure the confidentiality of this information. Especially in location sensors, due to the location sensitive nature of the transmitted data, the information should, under no circumstances, be accessed by unauthorized persons.

But, in addition to security applied to different type of sensors, the transmission of the collected data to the

aggregators have also to be done in a secure way. Transmission techniques like Secure Web Services, XML Encryption, XML Digital Signatures, XML Key Management System, Symbian, Windows Mobile, C# and .NET Security Frameworks, J2ME Security Frameworks, HTTP over TLS, SSL version 3.0, X.509 Public-Key Infrastructures and Cryptographic algorithms can be used for this aim.

We must make clear that in accordance with the type of sensors used and the under deployment scenario, the exact routing and aggregation scheme and the trade-off between safety and efficiency have to be considered, in order for the optimal solution to be selected.

3.2. Processing Layer

For sensors with limited processing power and communication bandwidth, processing raw data and sending large volumes of files is a problem. Sending large XML documents would cause many problems, such as energy consumption, routing problems and massive backlogs of information. To avoid this problem, we have designed the Processing Layer which is responsible for the processing of the stored data and their transformation to useful formats in order to make them meaningful for the end user. eXtensible Markup Language (XML) schemas can be used to publish formal descriptions of the sensor's capabilities, location, and interfaces. XML can be generated by the server after receiving the raw data.

Thus, an entity has to be developed which will be aggregating the data and transforming them in an XML (possibly a SensorML) format in order to be used later from the Semantic layer. This entity will support the integration of sensor data and the dynamic adaption of the system configuration during operation through a declarative XML based language. The user will have the ability to specify his preferred XML template, according to the data that have greater significance for his application. A module will also be available for transforming data from an XML template to another one with use of XSLT. Furthermore, if an existing encoding scheme is close to what is needed for a given application, it can be extended for that application. After the creation of the XML messages, data will be forwarded to the upper layer.

In this direction, an open-source tool that is nowadays available and can possibly be used in this layer is GSN [2]. In GSN, wrappers (in Java classes) are used to encapsulate the incoming data to the data model. Users can define and develop their own wrappers, in accordance with their data model. Then, the data streams are processed according to XML specification files.

3.3. Semantic Layer

The Semantic layer abstracts the processed outputs from the heterogeneous, low-level data sources such as sensors and feature extraction algorithms, thus enabling context capturing in varying conditions. Context annotation is configured through application-specific ontologies which can be plugged and it is automatically initiated without any further human intervention.

In order to incorporate Semantic Web technologies, the hereby presented architecture is based on a configurable modular approach [11]. As far as it concerns the knowledge manipulation task, we provide a set of core reusable distributed components for the real-time annotation of low level context features and their mapping to higher-level semantics which are directly exposed through suitable APIs for application development.

Once initiated, the Processing layer produces messages containing descriptions of the features captured by the sensors. Through these messages, knowledge is transferred to the main server, it is converted into semantic information via two distinct sets of rules, and a Knowledge Base (KB) is created where all the features of interest are gathered.

The Semantic layer architecture mainly comprises an exported Web Service interface, the Ontology Models, a set of Mapping and Semantic Rules with their corresponding actions/notifications and finally the external Reasoning Server. Each module is described analytically below.

3.3.1. Web Service interfacing module. The most important task of the Web Service module is message manipulation. What is required from messages is that they should be received from the Processing layer in any arbitrary well-formed XML document, without any additional constraints. When a message arrives, it is processed by the XML Mapping and Semantic Rules that the user has created in order to achieve the desirable behaviour.

3.3.2. Ontology models. The database model is stored using the Jena [8] [5] internal graph engine. The Jena framework has developed its own methodology for persisting ontology information. The ontological model is stored in triple statements which form the underlying graph of the model. The annotation is kept in the KB that is thereby created, separately from the incoming Sensor Web data. We also store the incoming XML files, in order to be transformed in another XML template if needed in the future.

3.3.3. Rules. The message processing phase comprises two distinct sets of rules. The XML Mapping Rules (Mapping Rules from now on) and the Semantic Rules, which are needed to provide syntactic and semantic homogeneity. The rules are formed according to the following event-condition-action pattern:

on *event* if *condition* then *action*

where the event in our case is the message arrival. The Mapping Rules fetch data from the XML message and store it into the ontology model in the form of ontology class individuals. They rely on the fact that for every XML element there is a unique XPath expression that retrieves its value. The Semantic Rules on the other hand perform modifications on the ontology model. Therefore, despite the common syntax, the condition and action differs in each case. The distinction of two variants of rules was inspired by the similar distinction between the RDF-only and the RDF-XML-combining subsets of RuleML [13] [9]. For instance, a Mapping Rule evaluated on an incoming XML message can state:

```
IF EXISTS /sensor/temperature/@value THEN
INSERT INDIVIDUAL IN CLASS Temperature AND
SET DATATYPE PROPERTY hasValue
/sensor/temperature/@value
```

```
IF EXISTS /sensor/humidity/@value THEN
INSERT INDIVIDUAL IN CLASS Humidity AND
SET DATATYPE PROPERTY hasValue
/sensor/humidity/@value
```

The Semantic Rule that will be applied consecutively will trigger an alert that will be based on the KB awareness of the world. The intelligence that is added and would not be subsistent without Semantic technologies is based on the fact that reasoning procedures can deduce implicit knowledge based on the current explicit facts.

```
IF DATATYPE PROPERTY IN CLASS
Temperature HAS VALUE GREATER THAN 40
AND DATATYPE PROPERTY IN CLASS Humidity
HAS VALUE GREATER THAN 0.3 THEN Alert
("Surveillance area under unusual conditions!")
```

3.3.4. The Reasoning Server. As stated in [3], a KB is the combination of an ontology and a reasoner, rendering the presence of the latter indispensable. There is a variety of available reasoners, commercial like RacerPro or OntoBroker, free of charge like KAON2 [15] and open-source like Pellet [19] and

FaCT++. All of them support DIG 2interoperability which is used by reasoners to exchange HTTP messages with programs that call them. Jena also supports the binding of an external reasoner, and provides a less adequate internal reasoner as well. The previously mentioned reasoners can function as stand-alone DIG servers and communicate with the middleware, leaving the reasoner choice up to the user.

4. Conclusions – Future Work

In this paper we have presented a modular architecture for deploying WSNs, in a manner which will ease end users to take advantage of the collected data and will facilitate developers to deploy useful applications, by exploiting the semantic web advances. We envision that such an architecture will add flexibility to the sensor world to form associations over the raw data, extract meaningful information and valuable results, and create specific management and notification rules, in accordance with the nature of each application.

Future work plans include the implementation of different scenarios combining several aggregation, security and processing methods and the evaluation of the discrete components of our proposed architecture. Special attention will be given to the performance of each layer entity in relation with the amount and the rate of the received data. Additionally, a study over the energy efficiency trade-off under several routing schemes and data aggregation architectures is going to be conducted;

5. References

- [1] Wireless Sensor Network Security: A Survey. Auerbach Publications, CRC Press, 2006.
- [2] K. Aberer, M. Hauswirth, and A. Salehi. Infrastructure for data processing in large-scale interconnected sensor networks. International Conference on Mobile Data Management, pages 198–205, 1-1 May 2007.
- [3] F. Baader and W. Nutt. The Description Logic Handbook. Cambridge University Press, 2002.
- [4] M. Balazinska, A. Deshpande, M. J. Franklin, P. B. Gibbons, J. Gray, M. Hansen, M. Liebhold, S. Nath, A. Szalay, and V. Tao. Data management in the worldwide sensor web. IEEE Pervasive Computing, 6(2):30–40, 2007.
- [5] J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson. Jena: Implementing the semantic web recommendations. Technical Report HPL-2003-146, Hewlett-Packard, Dec 2003.
- [6] M. Eid, R. Liscano, and A. El Saddik. A universal ontology for sensor networks data. CIMSAS 07., pages 59–62, 27-29 June 2007.
- [7] P. B. Gibbons, B. Karp, Y. Ke, S. Nath, and S. Seshan. Iris-Net: An architecture for a worldwide sensor web. IEEE Pervasive Computing, 2:22–33, Oct.-Dec. 2003.
- [8] L. Hewlett-Packard Development Company. Jena - a semantic web framework for java, 2006. <http://jena.sourceforge.net>.
- [9] D. Hirtle, H. Boley, B. Grosz, M. Kifer, M. Sintek, S. Tabet, and G. Wagner. Schema specification of ruleml 0.91. available online at <http://www.ruleml.org/0.91/>, 2006.
- [10] L. Kazatzopoulos, C. Delakouridis, G. F. Marias, and P. Georgiadiis. ihide: Hiding sources of information in wsns. In SECPERU '06, pages 41–48, Washington, DC, USA, 2006. IEEE Computer Society.
- [11] N. Konstantinou, E. Solidakis, S. Zoi, A. Zafeiropoulos, P. Stathopoulos, and N. Mitrou. Priamos: A middleware architecture for real-time semantic annotation of context features. In IE 07, Ulm, Germany, Sept. 2007.
- [12] M. Lewis, D. Cameron, S. Xie, and B. Arpinar. ES3N: A semantic approach to data management in sensor networks. In Semantic Sensor Networks Workshop (SSN06), Athens, Georgia, USA, Nov. 2006.
- [13] W. May, J. Alferes, and R. Amador. Rules and Rule Markup Languages for the Semantic Web, volume 3791 of Lecture Notes in Computer Science, chapter Active Rules in the Semantic Web: Dealing with Language Heterogeneity, pages 30–44. Springer Berlin / Heidelberg, 2005.
- [14] D. Moodley and I. Simonis. A new architecture for the sensor web: The SWAP framework. In Semantic Sensor Networks Workshop (SSN06), Athens, Georgia, USA, Nov. 2006.
- [15] B. Motik and U. Sattler. A Comparison of Reasoning Techniques for Querying Large Description Logic ABoxes. In M. Hermann and A. Voronkov, editors, LPAR 2006, volume 4246 of LNCS, pages 227–241, Phnom Penh, Cambodia, November 13–17 2006. Springer.
- [16] S. Rajeev, A. Ananda, C. M. Choon, and O. W. Tsang. Mobile, Wireless, and Sensor Networks - Technology, Applications, and Future Directions. John Wiley and Sons, 2006.
- [17] C. Reed, M. Botts, J. Davidson, G. Percivall, and F. Collins. Ogc sensor web enablement: overview and high level architecture. In AUTOTESTCON, Baltimore, MD, 2007.
- [18] D. Russomanno, C. Kothari, and O. Thomas. Sensor ontologies: from shallow to deep models. SSST '05., pages 107–112, 20-22 March 2005.
- [19] E. Sirin, B. Parsia, B. Grau, A. Kalyanpur, and Y. Katz. Pellet: A practical owl-dl reasoner. Journal of Web Semantics, 5(2):51–53, 2007.