In: Data Management in Semantic Web
Editors: Hai Jin, Zehua Lv
ISBN:

Chapter x

DATA MANAGEMENT IN SENSOR NETWORKS USING SEMANTIC WEB TECHNOLOGIES

Anastasios Zafeiropoulos*[*], *Dimitrios-Emmanuel Spanos, Stamatios Arkoulis, Nikolaos Konstantinou, Nikolas Mitrou
National Technical University of Athens,
9, Heroon Polytechneiou Str., 15773 Zografou, Athens, Greece

Abstract

The increasing availability of small-size sensor devices during the last few years and the large amount of data that they generate has led to the necessity for more efficient methods regarding data management. In this chapter, we review the techniques that are being used for data gathering and information management in sensor networks and the advantages that are provided through the proliferation of Semantic Web technologies. We present the current trends in the field of data management in sensor networks and propose a three-layer flexible architecture which intends to help developers as well as end users to take advantage of the full potential that modern sensor networks can offer. This architecture deals with issues regarding data aggregation, data enrichment and finally, data management and querying using Semantic Web technologies. 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.

1. Introduction

Sensor networks have attracted a lot of attention lately and have been increasingly adopted in a wide range of applications and diverse environments, from healthcare and traffic management to weather forecasting and satellite imaging. A vast amount of small, inexpensive, energy-efficient, and reliable sensors with wireless networking capabilities is available worldwide increasing the number of sensor network deployments [6]. Advanced networking capabilities enable transmission of sensory data through their connection to the

[*] Corresponding author: E-mail address: tzafeir@cn.ntua.gr, Phone: +302107722425

Internet or to specific gateways and provide remote access for management and configuration issues. The adoption of IPv6 also provides a huge address space for networking purposes in order to address the large sensor networks on a global scale while concurrently leads to the rapid development of many useful applications. Thus, it is not unreasonable to expect that in the near future, many segments of the networking world will be covered with sensor networks accessible via the Internet. Archived and real time sensor data will be available worldwide, accessible using standard protocols and application programming interfaces (APIs).

Nevertheless, as stated in [1], too much attention has been placed on the networking of distributed sensing while too little on tools to manage, analyze, and understand the collected data. In order to be able to exploit the data collected from the sensor network deployments, to map it to a suitable representation scheme, to extract meaningful information (e.g. events) from it and to increase interoperability and efficient cooperation among sensor nodes, we have to devise and apply appropriate techniques of data management. Towards this direction, data aggregation and processing have to be done in a way that renders it valuable to applications that receive stored or real time input and undertake specific actions. It is important to note that, special characteristics of sensor nodes, such as their resource constraints (low battery power, limited signal processing, limited computation and communication capabilities as well as small amount of memory) have to be considered while designing data management schemes.

Sensory data has to be collected and stored before being aggregated. Various techniques for data aggregation have been proposed in accordance with the type of the network and the imposed requirements [2] [3] [4]. However, aggregated data is raw data that has little meaning by itself. Hence, it is crucial to interpret it according to information that is relevant to the deployed applications. This will increase interoperability among different types of sensors as well as provide contextual information essential for situational knowledge in the sensor network. Moreover, appropriate methods of data processing can be proven helpful, especially in cases where data from many heterogeneous sources need to be compared and/or combined and different events have to be correlated. Towards this direction, the Open Geospatial Consortium (OGC) recently established the Sensor Web Enablement (SWE) initiative to address this aim by developing a suite of sensor related specifications, data models and Web services that will enable accessibility to and controllability of such data and services via the Web. The Sensor Web is a special type of Web-centric information infrastructure for collecting, modeling, storing, retrieving, sharing, manipulating, analyzing, and visualizing information about sensors and sensor observations of phenomena.

A promising technology that is able to address the technical challenges of extracting meaningful events and enabling interoperability among data from different sources, is the Semantic Web. The contribution of the Semantic Web is the semantic-level content annotation. Content 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 and the use of domain specific ontologies could enhance the use of knowledge extracted from the available information as well as add relationships between context data and application defined rules. In other words, Semantic Web can connect the sensory data to the features of their environment.

Taking into account the aforementioned considerations, in this chapter we present the basic characteristics of sensor networks and analyze in detail the problem of gathering, processing and intelligently exploiting real-time or stored streams of data produced by mostly heterogeneous sources of sensor nodes. Special emphasis has been given on the aspects of sensory data description, data transformation into meaningful information as well Semantic Web aided data processing that enables high level information extraction and sharing among different applications. Furthermore, we introduce a layered architecture that unifies the majority of the proposed approaches in the sensor data management research area and suggests a more straightforward recommendation for a complete solution for efficient data management in sensor networks.

2. Sensor Networks

2.1. Sensor Nodes: Functionality and Characteristics

A sensor node, also known as “mote”, was an idea introduced by the Smart Dust project [46] in early 00's (2001). Smart Dust was a promising research project that first studied and supported the design of autonomous sensing and communication micro-computing devices of size as small as a cubic millimeter (or the size of a “dust particle”). In other words, this project acted as the cornerstone for the development of today’s wireless sensor networks.

The key functionality of a modern sensor node, in addition to sensory data gathering, is the partial processing and transmission of the collected data to the neighbouring nodes or to some central facility. A modern node could be considered as a microscopic computer embedding all the units required for sensing, processing, communicating and storing sensory information, as well as power supply units able to support such operations. The most important units that are present in a sensor node are the following [48]:

- *the Processing Unit*, that is responsible not only for processing the collected data, but also for orchestrating the cooperation and synchronization of all other mote's units towards realizing the promised functionality. Its operation is most often supported by on-chip memory modules.
- *the Communication Unit*, also known as transceiver, that enables motes to communicate with each other for disseminating the gathered sensory data and aggregating them in the sink nodes (nodes with usually higher hardware specifications than simple sensor nodes). The two most popular technologies considered here are either the Radio Frequency (RF) one, where the unlicensed industrial, scientific and medical (ISM) spectrum band is worldwide and freely usable by anyone, or the Optical or Infrared (IR) one, where line-of-sight between communicating nodes is highly required – making communication extremely sensitive to the atmospheric conditions.
- *the Power Supply Unit*, that provides power for the operation of such tiny devices. A typical power source does not exceed the 0.5Ah under a voltage of 1.2V and is most commonly a battery or a capacitor. While operations like data sensing and processing consume some power, the communication between neighbouring nodes is proved to be the most energy-consuming task (by a factor of 1000, as compared to

the power consumed for taking a sample or performing a trivial aggregation operation [3]).

- *the Sensor Unit*, that is responsible for sensing the environment and measuring physical data. Sensors are sensitive electronic circuits turning the analog sensed signals into digital ones by using Analog-to-Digital converters. There is a large variety of sensors available today with the most popular of them being able to sense sounds, light, speed, acceleration, distance, position, angle, pressure, temperature, proximity, electric or magnetic fields, chemicals, and even weather-related signals. Such units must be able to provide the accuracy the supported application demands, while consuming the lowest possible energy.

Modern sensor nodes are required to be inexpensive, multifunctional, cooperative, microscopic, as well as able to cope efficiently with low power supplies and computational capacity. Consequently, contradictory issues are introduced regarding both motes' design and operation, with the most important of all being their limited lifetime, tightly related to their limited power resources, given that power supplies' replenishment, storing and harvesting is not a trivial issue until now. This is due to the fact that sometimes it is either extremely difficult, expensive, or even impossible to replenish motes' power supplies. This phenomenon is mainly due to the common practice followed today – in the case of terrestrial deployments – where sensor nodes are thrown randomly in mass, instead of being placed one by one in the sensor field according to special planning or engineering. Despite the fact that such an approach results in both pushing down the installation costs and increasing the overall deployment's flexibility, it leads most of the times to situations where the deployed sensors are hardly reachable, or even unreachable. As a result, redeploying a new sensor in a given area is rather preferable – in terms of cost, feasibility and provided effort – to getting in proximity with already deployed sensor nodes in order to replenish or fix them. Last but not least, a similar statement holds for the sensor nodes deployed either underwater or underground. The solution here would be to enable sensor nodes storing as much power as possible. However, their limited size does not allow for the use of large and heavy battery cells, while the use of cutting edge technology would violate their low manufacturing cost requirement. Alternatives such as harvesting power resources by exploiting solar energy, fuel cells or vibration are very popular today, but not so widely used yet in current deployments.

2.2. Sensor Networks Topologies

When a number of sensor nodes is clustered together, a special type of autonomic and power efficient network is formed, a so-called Wireless Sensor Network (WSN). WSNs are mainly consisting of the Sensor Nodes, the Sink Nodes that aggregate the measured data from a number of Sensor Nodes and the Gateway Nodes that interconnect the Sink Nodes with the network infrastructure (e.g. Internet) and route the traffic to proper destinations. There are cases where the Sink Nodes have embedded network interfaces for data forwarding and thus coincide with the Gateway Nodes. Regarding the topology of the sensor network, it may form either a single-hop network where each Sensor Node sends directly the data to the Sink Node through a star topology, or a multi-hop network where each Sensor Node relies on its neighbours to forward its sensory data to the respective Sink Node. Multi-hop networks may form a mesh, a serial or a tree topology, as it is shown in Figure 1. It is important to note that in most cases, the sensor nodes do not present mobility functionality.

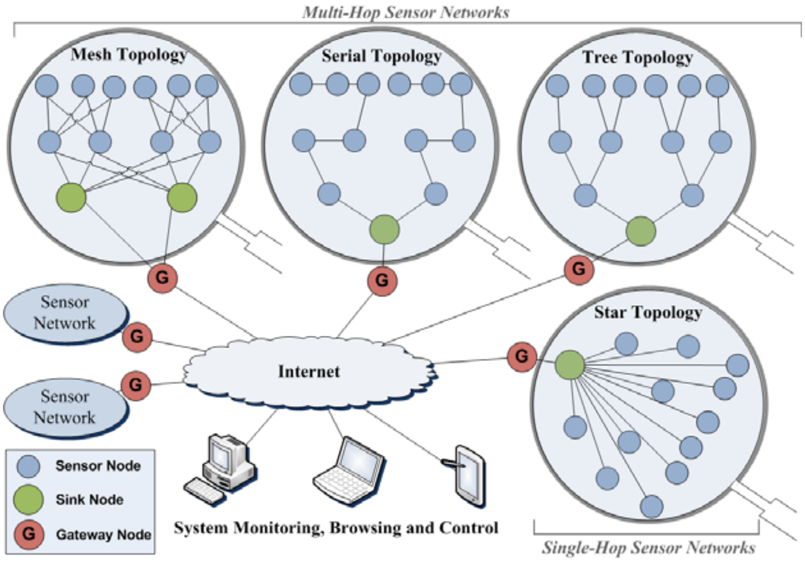

Figure 1. Sensor Network Topology.

Sensor networks carry several characteristics that make them unique. First of all, a sensor network's main objective is diametrically opposed to those of the traditional communication networks. The main concern and goal of sensor networks is their lifetime maximization instead of the provision of Quality of Service (QoS), a goal directly translatable into the need their overall power consumption minimization during operation. However, this is not always the case, since there are also sensor networks consisting of multimedia sensor nodes – able to cope with multimedia content (video, audio, and imaging) – where terms like QoS provisioning, high bandwidth demand, and small delay are more crucial than the power efficiency itself [47]. Furthermore, sensor networks must be fault tolerant, a term closely related to the required autonomicity of every single sensor node, since their underlying topologies are highly volatile and dynamic, as nodes continuously join or leave the network. New nodes may either be redeployed at a specific area to support the operation of an existing sensor network or have just overcome a critical problem that was obstructing their normal operation (e.g., environmental interference). As already mentioned before, a node's lifetime is proportional to its power supplies. When the latter drain out, the node becomes automatically inactive. Since such failures are very common in sensor networks, their overall operation should not be affected by them in any case. Thus, autonomic functionalities (self-healing, self-configuration, self-optimization, self-protection) should be developed in a sensor node. Fortunately, such techniques are fully supported by modern sensor networks, where millions of sensor nodes are deployed in extremely dense topologies; these techniques provide nodes with enough alternatives regarding their potential neighbors, cooperators, and routing paths. In such dense deployments, shorter communication distances have to be covered during the exchange of sensory data between Nodes/Gateways, thus leading to larger power savings during such an energy-consuming operation.

2.3. Application Areas

Sensor networks have been adopted in a wide set of scenarios and applications where proper data management can be deemed of high importance. Some of the application areas where deployments of sensor networks with advanced capabilities are popular are the following:

- *Health Monitoring*: Biometric sensors are usually used for collecting and monitoring data regarding patients, administrating issues in hospitals, provision of patient care as well as for supporting the operation of special chemical and biological measurement devices (e.g. blood pressure monitoring). The collected sensory data are also stored for historical reasons in order to be used for further survey on disease management and prognosis. In many cases, efficient representation and correlation of the acquired data enable doctors and students to extract useful conclusions.
- *Meteorology and Environment Observation*: Environmental sensors are used for weather forecasting, wildfire and pollution detection as well as for agricultural purposes. Special observation stations collect and transmit major parameters which are used in the procedure of decision making. For example, in agriculture, air temperature, relative humidity, precipitation and leaf wetness data are needed for applying disease prediction models, while soil moisture is crucial for proper irrigation decisions towards understanding the progress of water into the soil and the roots.
- *Industrial applications*: Different kind of sensors are deployed for serving industries including aerospace, construction, food processing, environmental & automotive. Applications are being developed for tracking of products and vehicles in transportation companies, satellite imaging, traffic management, monitoring the health of large structures such as office buildings and several other industry-specific fields.
- *Smart Homes*: Home automation applications are being developed in order to support intelligent artifacts and make the users' life more comfortable. Special sensors are attached to home appliances while the created sensor network can be managed or monitored by remote servers accessible via the Internet (e.g. user's office, police office and hospital etc). Sensor networks also play a significant role on facilitating assisted living for the elderly or persons in need of special care.
- *Defense (Military, Homeland Security)*: Sensors are also used for military purposes in order to detect and gain as much information as possible about enemy movements, explosions, and other phenomena of interest. Battlefield surveillance, reconnaissance (or scouting) of opposing forces, battle damage assessment and targeting are some of the fields where large sensor networks have been already deployed.

2.4. Sensor Web: Data and Services in a Sensor Network

The term Sensor Web is used by the Open Geospatial Consortium (OGC) for the description of a system that is comprised of diverse, location aware sensing devices that report data through the Web. In a Sensor Web, entire networks can be seen as single interconnected nodes that communicate via the Internet and can be controlled and accessed through a web interface. Sensor Web focuses on the sharing of information among nodes, their proper interpretation and their cooperation as a whole, in order to sense and respond to changes of their environment and extract knowledge. Hence, one could say that the process of managing the available data is not just a secondary process simply enhancing the functionality of a Sensor Web, but rather the reason of existence of the latter. According to [3], data management can be defined as the task of collecting data from sensors (sink or gateway nodes), storing data in the network, and efficiently delivering data to the end users.

One of the most appealing and popular approaches in sensor data management is to regard the sensor network as a distributed database system. Data management on distributed database environments is a far more mature research field than the rapidly evolving sensor networking field, thus it sounds tempting to use the experience gained in the former in order to deal with open issues in the latter. However, this parallelism that regards sensor nodes as databases, is obviously, not entirely realistic, since a sensor node does not have the computational capabilities or the energy sufficiency that a database management system (DBMS) has. In other words, the technological constraints, mentioned in the previous section, should be taken into account towards devising efficient data management methods. Frequent failures of sensor nodes, unreliable radio communication and tradeoff between low energy consumption and network efficiency have to be considered when designing redundant and robust data management approaches, ensuring the proper functionality of a sensor network. Furthermore, the usually large number of sensor nodes in a Sensor Web system creates the demand for scalable solutions of data management and thus promotes local, asynchronous approaches that use short-range transmissions among the interconnected nodes. Thus, although data management in sensor networks borrows heavily from research on distributed databases, extra constraints are imposed due to innate characteristics of sensor nodes.

Primarily, the acquired data have to be stored taking into account methods for low energy consumption. Data storage can be either external (all data are collected on a central infrastructure), local (every node stores its data locally) or data-centric (a certain category of data is stored to a predefined node). External storage is not considered as a viable solution, because of the high energy cost of data transmission from each sensor node to the central infrastructure. Local storage overcomes this drawback, since every node stores only self-generated data. The option of local storage is also referred to as Data-Centric Routing (DCR), where a routing algorithm is needed in order to answer a query or to perform an aggregation, focusing on minimizing the cost of communication between sensor nodes. In early examples of DCR [19, 20], queries are flooded throughout the entire network and on the same time, a routing tree is built. Then, data from the individual sensor nodes or aggregated data are returned, following the query tree path to the node that initiated the query or the request. This procedure is repeated for every different query and therefore, is efficient only for queries that are executed on a continuous basis, e.g. query tracking or repeated computation of aggregates for some sensors. The third option of following a data storage policy is the most recent one, known in the literature as Data-Centric Storage (DCS). In this approach, data and events with common properties or the same attribute (e.g. temperature readings) are stored to specific nodes, regardless of where the data has been generated. Example methods include DIM [21] and GHT [22], which, by using distributed data structures (an index and a hash table respectively) avoid query flooding, and instead, direct the query to the node that is delegated to store the relevant data. It is obvious that the choice of data storage strategy affects the way that queries are answered, the data flow among sensor nodes and the ability of the Sensor Web to answer questions regarding both historical and present data.

Another aspect of data management is data aggregation, a common operation in sensor networks regardless of the deployment scenario. Data aggregation can be defined as the collection of data originating from a number of distributed sensor nodes to a sink node, offering the advantage of less data being transferred across the network. Sometimes, in a sensor network, there are two or more proximate sensors that measure the same phenomenon; in this case, an aggregated view (e.g. the mean value) of these sensors' observations is equally or even more desirable than the individual readings, due to the possible redundancy of the measurements. Thus, it becomes clear that there are at least two issues to be taken into account: a) the selection of the sink nodes, based on their computational capabilities (unless the final processing takes place in a centralized server in the network) and b) the selection of the most power-efficient strategy to be followed for the aggregation procedure. Of course, these two issues are correlated and the selection of the so-called "leader" node should not ignore the goal of minimum energy consumption. There are several criteria that could be used for the election of the leader node; for example, the leader node could be the node with the greatest remaining energy supply or the node that will minimize the exchange of messages across the network based on its topology. Furthermore, more sophisticated election algorithms could be devised [24].

Computational procedures usually need to be performed on the aggregated data. Two alternative strategies have been widely used for computational issues. In the more simplistic centralized approach, all the nodes send their data to the central node, while in the distributed approach, every node performs a part of the total processing and passes on to the neighbouring nodes the result of this partial computation along the path of a routing tree built for this purpose. In the distributed approach, each sensor node must have computational capabilities in order to be able to process the available data. Synchronization issues arise as well, since nodes have to know if they are expecting any data from their children in the routing tree in order to pass on the partial result to their parent. In most cases, the sensor node is not aware if there is a big communication delay with another node or if an existing link is broken. Hence, the problem here is how long the parent node has to wait: either a fixed time interval [23] or until it receives a notification from all of its children that have not sent any data yet [24]. Data aggregation methods are being evaluated on the basis of accuracy (the difference between the aggregated value that reaches the sink and the actual value), completeness (the percentage of readings included in the aggregation computation), latency and the message overhead that they introduce.

Data management includes also delivery of data generated by sensor nodes to the users that request it. Query answering is tightly connected to data storage and aggregation, which we have already analyzed. In fact, data aggregation can be considered as a special case of query answering, since aggregation queries are simply one case of possible queries that a user may submit to the sensor network. Other kinds of queries are one-shot queries, asking for a value of a specific sensor at a specific time, event-based queries, triggered by an event defined in the query and lifetime-based queries, specified to run for a larger time period. Of course, the goal here is the same: execute queries with as little energy consumption as possible. Depending on the type of query issued and the type of sensor, sampling rate can be adjusted accordingly in order to save energy. For example, in the case of PIR (Passive InfraRed) motion detectors, sensor data become quickly outdated, and therefore, in a monitoring scenario, the query retrieving these sensors' values has to be executed frequently. On the contrary, temperature or other environmental variables exhibit small variance and approximate results are satisfactory, allowing for caching previous results and lowering the query update rate. Efforts have been made to reduce energy consumption in all stages of query answering: query optimization, dissemination and execution [23]. However, there is a tradeoff between energy-efficient and complete query answering schemes; in other words, the more energy-saving a query answering approach is, the less correct and accurate the answer will be. Typically, the process of query answering is based on the formulation of a routing tree, along which the query is first disseminated and then data is returned, following the opposite direction. From this point on, approaches vary: some simply choose the most energy-efficient query plan, based on the form of the query and the topology of the network [25], some apply extensively in-network aggregation constructing appropriate routing trees [26], while others modify accordingly the sampling rate of the sensors [23].

Another feature of sensor networks that poses further challenges to query answering is the fact that, unlike database management systems that follow a pull-based model for data dissemination, sensor networks usually follow a push-based model, where new data is constantly generated and may trigger event-based queries. This differs radically from the usual scenario of a user (or a node) submitting a query and having the answer returned to him. In this case, when streams of data (readings) are being pushed in the Sensor Web, continuous queries must react to this data and then, answers are submitted to the interested user. Several architectures have been proposed in order to deal with push-based data stream processing, such as Aurora [27], TelegraphCQ [28] and Borealis [29].

So far, we have assumed that nodes in a Sensor Web are stationary and the main challenge is to keep the energy consumption as low as possible. But, there are cases where sensor nodes are mobile and thus pose further challenges to data management. For instance, the external data storage is not an option for mobile sensor networks, since mobile nodes may not always be able to access the sink node, making in-network storage the only choice. Similarly, approaches for data aggregation and query answering must be adaptive because of the constantly changing topology.

All of the previously mentioned issues have emerged due to the inherent restrictions of sensor networks. However, with the introduction of the Sensor Web notion, challenges and demands increase. Further interoperability and flexibility play central roles in the Sensor Web vision, since different kinds of sensors must be linked and communicate with each other, while at the same time it should be easy for new sensors to be added in an existing Sensor Web. For this purpose, the Open Geospatial Consortium (OGC) has developed and maintains, in the context of its Sensor Web Enablement (SWE) initiative [7], a series of standards. These standards include, on one hand, markup languages defining a vocabulary for the common understanding and encoding of observations, measurements, sensor processes and properties, such as Observations & Measurements Schema (O&M), Sensor Model Language (SensorML) and Transducer Markup Language (TransducerML) and on the other hand, web service interfaces for the request and retrieval of sensor measurements, acquisition of measurements planning and subscription to event alerts, such as Sensor Observations Service (SOS), Sensor Planning Service (SPS), Sensor Alert Service (SAS) and Web Notification Services (WNS). Obviously, to achieve cooperation among different sensor networks, agreement on some common representation scheme must be made; it is believed that OGC standards will play the role of the bond that glues together all of the Sensor Web components.

3. Knowledge Management in Sensor Networks

As stated earlier, the rapid development and deployment of sensor technology involves many different types of sensors, both remote and in-situ, with diverse capabilities. However, the lack of integration and communication between the deployed sensor networks often isolates important data streams and intensifies the existing problem of too much data and not enough knowledge. 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.

Current efforts at the OGC Sensor Web Enablement (SWE) aim at providing interoperability at the service interface and message encoding levels. Sensor Web Enablement presents many opportunities for adding a real-time sensor dimension to the Internet and the Web. It is focused on developing standards to enable the discovery, exchange, and processing of sensor observations. The functionality that OGC aims to supply a Sensor Web with, includes discovery of sensor systems, determination of a sensor's capabilities and quality of measurements, access to sensor parameters that automatically allow software to process and geo-locate observations, retrieval of real-time or time-series observations, tasking of sensors to acquire observations of interest and subscription to and publishing of alerts to be issued by sensors or sensor services based upon certain criteria [7].

However, these Sensor Web Enablement functionalities are not enough, when it comes to real-life applications, where requirements are extended beyond simple "readings-retrieval" queries. In such cases, the need for well-defined semantics that will enhance the sensors by providing situation awareness is evident. Technologies and standards issued by the World Wide Web Consortium (W3C) will be used in this context to implement the Semantic Sensor Web (SSW) vision [8], an extension of the Sensor Web, where sensor nodes will be able to discover their respective capabilities and exchange and process data automatically without human intervention. Components playing a key role in Semantic Sensor Web are ontologies, semantic annotation, query languages and rule languages.

Ontologies are formal representations of a domain that can serve as dictionaries containing the definitions of all concepts (phenomena, temporal and spatial concepts) used throughout the Sensor Web. Semantic annotation languages, such as Resource Description Framework - in - attributes (RDFa), do just what their name suggests: they enrich existing content with semantic information. Since Sensor Web Enablement standards are XML-based, RDFa can be used for annotating sensors' measurements and observations. Then, standard reasoning services can be applied to produce inferences on existing facts and spawn new knowledge. Rules can be defined using SWRL (Semantic Web Rule Language) and additional knowledge can be extracted by applying rule-based reasoning. Moreover, complex queries written in SPARQL Query Language for RDF - a W3C recommendation (or equivalently, a standard for the Web) - can be submitted to the Sensor Web for meaningful knowledge extraction and not just for simple retrieval of sensor readings. The previously mentioned Semantic Web technologies are the most prominent ones, but there are several others as well that could be proven useful in the context of Semantic Sensor Web. The application of these technologies will transform the Sensor Web Enablement service standards to Semantic Web Service interfaces, enabling sensor nodes to act as autonomous agents being able to discover neighbouring nodes and communicate with each other.

3.1. Current Approaches

In this section we present the state of the art in the field of knowledge management in sensor networks. Many approaches are available today for managing sensor networks, regarding especially the aggregation and processing of data, and several architectures have been proposed that provide services to the end user through the exploitation of the collected data. Existing approaches 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. They usually provide a software infrastructure that permits users to query globally distributed collections of high bitrate sensors' data powerfully and efficiently. Following this approach, the SWAP framework [11] proposes a three tier architecture comprising a sensor, a knowledge and a decision layer, each one of them consisting 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. In the same direction, IrisNet [10] 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 mean to answer user queries.

Recently proposed approaches go one step further and apply Semantic Web technologies on top of a sensor network in order to support the services provided in the existing and the newly deployed sensor networks. These technologies allow the sensor data to be understood and processed in a meaningful way by a variety of applications with different purposes. Ontologies are used for the definition and the description of the semantics of the sensor data. Such an approach is the ES3N architecture [9] that develops an ontology-based storage mechanism for sensor observations that 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. Another proposed architecture that exploits Semantic Web technologies is the SWASN architecture [17] where mechanisms for context aware processing of sensor data in pervasive communications scenarios are defined and the Jena API is used for sensor data processing to query sensor data and extract meaningful information through inference. Finally, Priamos [12] is a middleware architecture for automated, real-time, unsupervised annotation of low-level context features and their mapping to high-level semantics. It 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.

Going one step further, Semantic Sensor Web (SSW) is proposed as a framework for providing enhanced meaning for sensor observations so as to enable situation awareness [8]. This is accomplished through the addition of semantic annotations to existing standard sensor languages of the Sensor Web Enablement. These annotations provide more meaningful descriptions and enhanced access to sensor data in comparison with the Sensor Web Enablement alone, and they act as a linking mechanism to bridge the gap between the primarily syntactic XML-based metadata standards of the Sensor Web Enablement and the RDF/OWL-based vocabularies driving the Semantic Web.

As we have described earlier, the semantic representation of sensory data is significant because ontologies specify the important concepts in a domain of interest and their relationships, thus formalizing knowledge about the specific domain. In association with semantic annotation, ontologies and rules play an important role in the Semantic Sensor Web vision for interoperability, analysis, and reasoning over heterogeneous multimodal sensor data. Several ontologies for describing entities and relationships in sensor networks have already been designed. A universal ontology is designed for describing concepts and relationships of the sensor network units and data [13]. The source for collecting commonly used terms in sensor domain and the taxonomic class diagram that forms the foundation of the ontology was the IEEE 1451.4 smart transducers template description language [18]. In a more practical approach, the development of comprehensive sensor ontologies is based upon deep knowledge models rather than capturing only superficial sensor attributes [14]. Thus, the OntoSensor ontology has been proposed, providing formal definitions of the concepts and relations in sensor networks, being influenced from SensorML and extending concepts from the IEEE SUMO ontology [13]. It presents a practical approach to building a sensor knowledge repository aiming to serve as a component in comprehensive applications that include advanced inference mechanisms, which can be used for synergistic fusion of heterogeneous data. Furthermore, there are some service-oriented approaches that describe sensor ontologies which enable service-oriented services in future ubiquitous computing [15]. The main sources for collecting commonly used terms in the service domain are the Geography Markup Language (GML), SensorML, SUMO and OntoSensor.

Finally, it is important to note that environmental sensor data captured by an ontology can be combined with a rule-based system that can reason over the ontology instances creating alarm-type of objects if certain conditions are met [16]. The rules are fired based on data obtained from sensors in real-time and classified as instances of fundamental ontologies. The rule-based system is used as an advisor in the complex process of decision making, applying corrective measures when needed. The decision combines the real-time data with a-priori sensor data stored in various warehouses. The platform controls also the whole environment in order to make the user aware of any glitches in the functionality of the whole system.

3.2. A Unifying Generic Architecture for Sensor Data Management

However varied and divergent some of the above approaches might seem, they share enough common traits allowing us to propose, in this section, a generic scheme for efficient sensor data management, combining as many desirable features and interesting aspects as possible. This generic scheme is applicable to a wide range of different categories of sensor networks and deployed applications. After completing a review of the already deployed systems, we can infer that the downside of the majority of the proposed approaches are the limitations regarding the size of the sensor network, the amount of data 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. The limited presence of semantic context annotation and ontological descriptions for high-level rules and queries restricts the potential of end users to exploit the acquired information.

The architecture we describe tries to tackle with these issues and provide a flexible and modular scheme for easily deploying and managing large scale sensor networks, consisting of heterogeneous data sources. It consists of three layers (see Figure 2): the Data Layer for data discovery, collection and aggregation, the Processing Layer for integration and processing of the aggregated data and the Semantic Layer for the addition of context annotation and ontological descriptions. The advantage of such a scheme is the fact that each layer can be considered independent from the others and consequently, layer-specific decisions about the implementation and strategies followed in one layer do not affect other layers. This allows for harvesting and reusing previous work on traditional data management in sensor networks, where the emphasis has been given on dealing with the limited energy restriction.

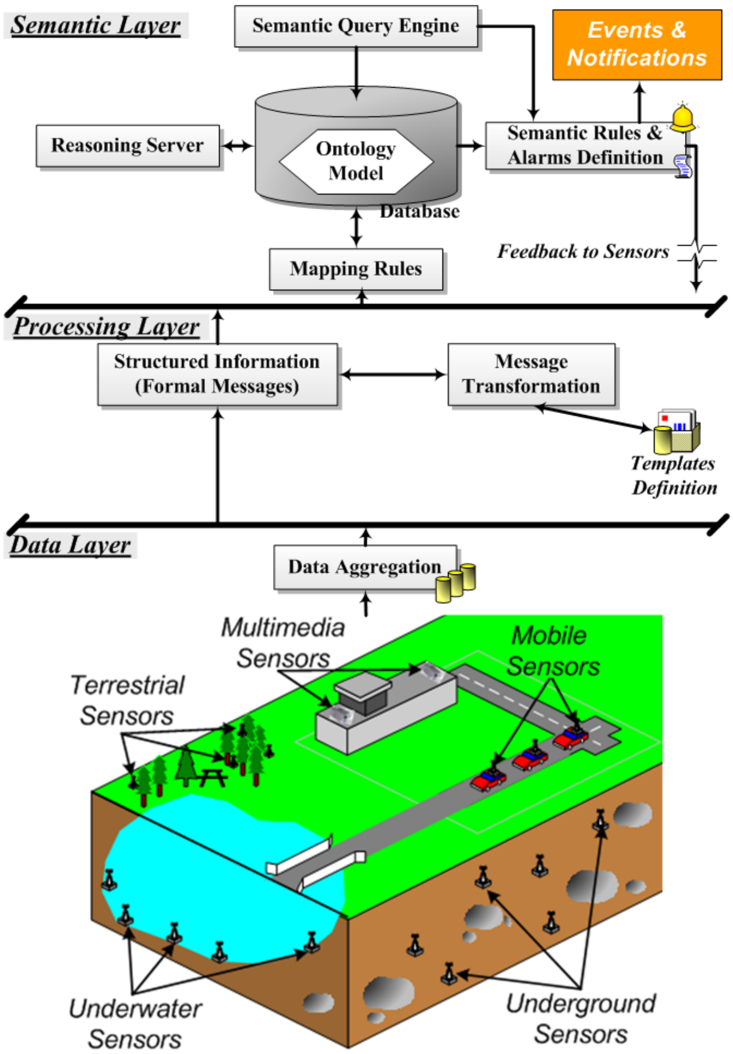

Figure 2. Abstract Overall Architecture.

In more detail, in the Data Layer, this architecture does not reject past data management techniques, but builds on the most successful ones. Different topologies and data aggregation techniques are supported, providing flexibility to administrators to select the most suitable topology and aggregation scheme for optimizing energy consumption and bandwidth utilization. The Processing Layer exploits all the recent advances in data processing and provides interfaces for selection of different templates according to the user needs and the transformation to new templates when required (e.g. new sensors are introduced in the sensor network that provide extra measurements). Thus, aggregated data is always represented in the most suitable format that enhances their meaning. The layered scheme we introduce incorporates even traditional approaches that do not use any semantics and are not context aware; in this case, we can assume that the Semantic Layer is simply omitted. The existence of the Semantic Layer is purely optional, depending on whether the addition of semantics improves considerably the functionality of the entire application or not. But, the use of Semantic Web technologies for data management in sensor networks gives to developers the opportunity to create applications that can sense the environment and deploy, identify the relationships among the existing entities and provide interfaces for enhanced query and reasoning within the sensor domain. Furthermore, we should note that modularity is a key aspect of this architecture, since it is indispensable to decouple the collection from the processing and semantic enrichment of data. Hence, when trying to build a sensor-based application that will exploit semantic web technologies, an engineer is not compelled to design everything from scratch, but can simply deal with the construction of efficient algorithms and implementation details in the top layers.

3.2.1. Data Layer

This layer handles raw sensor data discovery, collection and aggregation in a central entity. Efficient data aggregation is crucial for reducing communication cost, thereby extending the lifetime of sensor networks. As we have described in Section 2.2 different kinds of topology may be present in a sensor network. Based on the topology of the network, the location of sources and the aggregation function, an optimal aggregation structure can be constructed. Optimal aggregation can be defined in terms of total energy consumption, bandwidth utilization and delay for transporting the collected information from simple nodes to the sink nodes. Data gathering can be realized following structured or structure-free approaches [2]:

- Structured approaches are suited for data gathering applications where the sensor nodes are following a specific strategy for forwarding the data to the sink nodes. Due to the unchanging traffic pattern, structured aggregation techniques incur low maintenance overhead and are therefore suited for such applications. But, in case of dynamic environments, the overhead of construction and maintenance of the structure may outweigh the benefits of data aggregation. Furthermore, structured approaches are sensitive to the delay imposed from the intermediate nodes, the frequency of the data transmission and the size of the sensor network. The central entity is 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 structure-free approaches, there is no predefined structure and routing decisions for the efficient aggregation of packets need to be made on-the-fly. As nodes do not explicitly know their upstream nodes, they cannot wait on data from any particular node before forwarding their own data. These approaches can be applied in dynamic environments and in ad-hoc sensor networks where nodes continuously join and leave the network and thus a predefined static routing and aggregation scheme will possibly be unsuitable.

In addition to raw data aggregation, security awareness is an important aspect that is considered due to the reason that sensor networks often store, process and interact with sensitive data in hostile and unattended environments. Security in sensor networks 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 [42]. In cases where the location of important sensor-observed aspects must be hidden, anonymity mechanisms [43] 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.

We must make clear that depending on the type of sensors used and the deployment scenario, the exact routing and aggregation scheme and the trade-off between safety and efficiency have to be taken into account for the optimal solution to be selected.

3.2.2. Processing Layer

Due to the raw nature of sensory data and the fact that it cannot provide us with high-level information extraction, several XML-based models are being used in order to interpret it. This will leverage its usability, allow further processing and finally make it meaningful for the end user. Proper processing is necessary, especially in cases of aggregation of data from many heterogeneous sources and the need for discovery of possible correlations among the aggregated data. Furthermore, the processed data can be distributed to other network devices (e.g. PDAs) without the need for sensor-specific software. Different XML templates can interpret in a different way the sensory data according to the application related view. The aggregated data has to be processed and integrated in a manner that shortens the data exchanging transactions. Integrating the data and transforming it into an XML (possibly a SensorML) format makes it meaningful for the end user.

Initially, the Processing Layer integrates the bulk of the incoming data. It is not necessary neither optimal, in certain cases, to maintain the total amount of data. Consider for instance a sensor network consisting of some dozens of sensors measuring the temperature over a field. While keeping track of the temperature levels is useful, processing every single datum originating from every single sensor is not needed. Such practice would overload the network, augment its maintenance needs and consequently decrease its autonomicity. Moreover, the volume of the archived information would soon require a substantial storage capacity. Aggregated reports (such as the maximum or an average of the values reported) may be sufficient to describe the conditions that are present in the area of interest.

Subsequently, the integrated information collected by the sensors has to be forwarded to the upper Semantic Layer. In order for this to be achieved, the information needs to be encapsulated in messages suitable for further machine processing. For instance, XML-based languages such as OGC’s SensorML, Geography Markup Language (GML), Observations and Measurement (O&M) can stand up to the challenge. It is important to note that SensorML is currently encoded in XML Schema, but the models and encoding patterns for SensorML follow Semantic Web concepts of Object-Association-Object. Therefore, SensorML models could easily be transformed to a more “semantic” representation form. Furthermore, SensorML makes extensive use of soft-typing and linking to online dictionaries for definition of parameters and terms.

The process of creating a message with information originating from the observations of the sensor network constitutes a substantial computational burden for conventional sensor nodes, which means that this processing should take place at a central infrastructure, i.e. a server that administers the sensor network. The message templates and their transformation constitute a common agreement on the messages interchanged in the network. The template definition and its corresponding transformation can be thought of as an XML Schema Definition (XSD) and an Extensible Stylesheet Language Transformation (XSLT) in order to produce an XML message. It is needed that the user will have the ability to specify his preferred message templates in order to ensure adaptability to the system's requirements, according to the data that has greater significance for his application. Thus, after the creation of the XML messages, data has to be forwarded to the Semantic Layer.

3.2.3. Semantic Layer

The Semantic Layer abstracts the processed outputs from the heterogeneous, low-level data sources such as sensors and feature extraction algorithms, combined with metadata, thus enabling context capturing in varying conditions. Context annotation is configured through application-specific ontologies and it can be automatically initiated without any further human intervention. It must be noted that the Semantic Layer is not an indispensable part of a sensor network architecture, in the same way that semantics do not need necessarily to be part of systems. However, the benefits provided by a Semantic Layer justify its existence, regardless to the additional computational burden. As we analyze in this section, Semantic Web technologies allow for complex context descriptions, reusable rule definitions, concept satisfiability and consistency, and finally, integration with third-party systems and services. A Semantic Layer can be viewed as consisting of the following modules, as depicted in Figure 2.

Rules: First, the incoming messages must be mapped to ontology concepts. The use of rules is essential in depicting the desired behaviour of context-aware systems. Since the model-theoretic background of the OWL language is based on Description Logics systems that are a subset of the First Order Logic, the designed model has fully defined semantics and, also, Horn clauses can be formed upon it. These clauses can be seen as rules that predefine the desired intelligence in the system's behavior.

In general, two distinct sets of rules can be applied: one containing rules that specify how sensor measurements represented in an XML-based format will be mapped to a selected ontology and another set of rules deriving new facts, actions or alerts based on existing facts and the underlying knowledge represented by the ontology. The first set can be referred to as Mapping Rules and the second one as Semantic Rules. The rules are formed according to the following event-condition-action (ECA [31]) pattern:

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

where the event in a sensor network is a message arrival indicating a new available measurement. Mapping Rules can fetch data from the XML-like message and store it into the ontology model in the form of class individuals. They can be perceived as the necessary step bridging the gap between semi-structured data presented in an XML form and ontological models. Semantic Rules, on the other hand, can perform modifications, when needed, solely on the ontology model. This set of rules depend on the semantics of the specific domain or deployment scenario and involves high-level concepts that are meaningful to humans e.g. "when a certain area under observation is "too hot", open the ventilating system". Of course, the "too hot" conclusion will probably be inferred from a set of current observations coupled with the knowledge stored in the ontology.

Ontology model: Contemporary Semantic Web practices dictate that the ontological model is stored in a relational database in triples (statements of the form Subject, Property, Object) and form the underlying graph of the model. The model can be stored as an RDF graph using one of the various so called triple-store implementations available (3Store [32], Corese [33], Sesame [34], Openlink's Virtuoso [44] to name a few).

In order to guarantee system's scalability, sampling techniques or caching for future processing can be used. For instance, an ontology model that handles the incoming messages, i.e. a temporary ontology model, can be kept in a different database from the persistent ontology model. The tradeoff for this approach is that the reasoning that takes place for every new message is aware of the facts that are stored in the temporary ontology model. Additionally, scheduled maintenance can be responsible for migrating the facts from the temporary ontology to the persistent storage. This scheduled task can take place either synchronously or asynchronously. The former case indicates that the migration can be triggered by an incoming message while the latter that the migration schedule can be running as a background process according to specified time intervals.

We also must note that the use of several Semantic Web vocabularies is desirable and should be encouraged in order to allow unambiguous definitions of the concepts involved in any application. Vocabularies, such as the Dublin Core Metadata Initiative for digital content, DBPedia [45] for reference to Wikipedia's entries, the Friend Of A Friend network or the Creative Commons for licensing purposes, provide the means for effective semantic interoperability between applications.

As far as the ontology content is concerned, we should keep in mind that the ontology should describe the system's context. According to [35], context means situational information and involves any information that can be used to characterize the situation of an entity. An entity can be a person, a place, or an object that is considered relevant to the interaction between a user and a system, including the user and the system themselves. Therefore, specific ontology authoring can be inspired by approaches to describe context, such as the approach presented in [30] where the authors provide a rough categorization of context into four classes: location, time, activity and identity. The choice of an authoring environment is up to the user. According to [37] the most used ontology authoring environments are Protégé, SWOOP and OntoEdit.

Reasoning Server: As stated in [36], a Knowledge Base is the combination of an ontology and a software engine, also known as reasoner, capable of inferring facts from the facts explicitly defined in the ontology (sometimes referred to as axioms). The presence of a reasoner is indispensable, since it is the module that spawns new knowledge. Furthermore, a reasoner inspects an ontology model and can check its consistency, satisfiability and classification of its concepts [38], leading to the existence of a "healthy" and robust context model that does not contain any disaccords among its term definitions.

Reasoning is an important and active field of research, investigating the tradeoff between the expressiveness of ontology definition languages and the computational complexity of the reasoning procedure, as well as the discovery of efficient reasoning algorithms applicable to practical situations. There is a variety of available reasoners, commercial ones like RacerPro or OntoBroker, free of charge like KAON2 [39] and open-source like Pellet [40] and FaCT++ [41], which have different features and performance characteristics according to the application specific needs, e.g. in case of voluminous knowledge bases, a reasoner that scales well to millions of facts should be used. All of these reasoning servers can function stand-alone and communicate via HTTP with the deployed system, leaving the reasoner choice up to the user.

Semantic Queries: The Semantic Layer should also provide the user with a way to ask and derive information from the underlying Knowledge Base, similarly to relational database queries. Hence, an important component of this layer is a semantic query interface that will allow for the creation and execution of intelligent queries, based on the semantics of the stored knowledge, in contrast to the conventional keyword-based queries such as in SQL for relational databases or XQuery for XML. SPARQL, a W3C recommendation since 2007 as a query language for the Semantic Web, is the practical choice for querying RDF graphs and a SPARQL endpoint can essentially serve as the gateway to access the stored knowledge from remote systems.

3.2.4. Use Case Scenario

This section illustrates a scenario where all the previously described features come into play. Arrays of sensors can be placed inside art exhibitions or museums for monitoring purposes. In the current example, we consider a museum where temperature, humidity, light and passive infrared sensors are placed in proximity to each exhibit: paintings, sculptures and other, possibly sensitive in environmental changes, artifacts. These sensors have a fixed known location, forming a static sensor network. Location sensors (such as Bluetooth Tags [49]) are also located in each room and are used for positioning purposes. Such a deployment may exist either in open air or, for the purpose of our example, in an indoor environment.

The collected data is aggregated on several sink nodes connected to specific gateways which forward the measured data to a central repository either on the museum or in a remote site. Then, the aggregated data is processed and mapped to an Observations and Measurement (O&M) template, which optimally encodes sensor observations. Safety intervals are denoted explicitly for every exhibit – in the form of rules – and stored in an ontology that contains additional information about the artifacts and their creators and describes adequately the important entities of the current scenario as well as the relationships among them.

Each exhibit has different properties and different temperature, light and humidity levels are required for its optimal preservation. Specific rules can be defined for triggering the desired alarms, allowing application developers to provide location-aware services in the museum. Museum's personnel can move around the museum with a mobile device, recognize the current position through proper localization and check whether the existing artifacts in this room are under optimal conditions or not. Of course, when sensors deployed near an exhibit report a temperature or humidity value that is outside the respective safety interval, an alarm fires off automatically and an action selected by the system administrator is performed: either a direct notification to the museum personnel or the appropriate notification to the museum's Heating, Ventilating, and Air Conditioning (HVAC) system for automatic temperature, light or humidity adjustment. The reasoning server is responsible to reason over the acquired data towards providing the required information to the administrators.

The combination of sensor readings and the ontology information can be used for precautionary reasons as well. A SPARQL endpoint provides access to the system's ontology and can answer complex questions, such as:

"Return the positions and the names of all the 18th century landscape paintings in this room, created by a Dutch artist, whose temperature, or light or humidity readings are higher than 80% of the maximum allowed values" or

"Return the most suitable room for installation of a new artifact according to its category and its prerequisite conditions"

Furthermore, the passive infrared sensors detect possible motion of the exhibits and thus, can be used for security reasons to inform when someone touches or tries to remove an artifact or enter a banned region near an artifact, and in this case, an appropriate notification is also being sent to the museum personnel. Of course, more scenarios of usage can be thought of, depending on the defined rules and actions residing in the semantic layer of the system. Finally, the already stored data can be used for offline analysis and useful statistics extraction for each area.

4. Conclusions – Open Issues

In this chapter we have presented the available techniques that are being used for data management in sensor networks. Sensor network characteristics, possible topologies, data aggregation and data querying schemes are described in detail. The use of Semantic Web technologies for extracting meaningful events from the aggregated data is investigated. A generic modular architecture for knowledge management in sensor networks that consists of the Data, the Processing and the Semantic Layer is presented. We envision that such an architecture will add flexibility to the sensor world to form associations over the raw data, extract information and valuable results, and create specific management and notification rules, in accordance with the nature of each application. It will also facilitate developers to create new services for the end users and to provide them with context-aware information.

Open issues include a) the energy efficiency trade-off under several routing schemes and data aggregation architectures: selection of the suitable gathering strategy under different topologies can optimize use of resources in the sensing field, b) the proper dissemination of the available information among the sensor nodes: collaborative data processing can minimize the amount of data transferred and information exchange can help sensor nodes improve their self-functionalities (e.g. self-configuration), c) the analysis of mechanisms for the establishment of accurate synchronization: participating nodes will be synchronized either by exchanging messages with other nodes or by communicating with a central entity in order to acquire a common notion of time, d) the implementation of different scenarios combining several aggregation, security and processing methods and the evaluation of the discrete components of the proposed architecture. Special attention also has to be given to the performance of each layer entity depending on the amount and the rate of the received data.

References

[1] M. Balazinska, A. Deshpande, M. J. Franklin, P. B. Gibbons, J. Gray, M. Hansen, M. Liebhold, S. Nath, A. Szalay, V. Tao, Data Management in the Worldwide Sensor Web, *IEEE Pervasive Computing*, **6(2)**, p. 30-40 (2007).

[2] K.W. Fan, S. Liu, P. Sinha, Structure-Free Data Aggregation in Sensor Networks, *IEEE Transactions on Mobile Computing*, **6(8)**, p.929-942 (2007).

[3] V. Cantoni, L. Lombardi, P. Lombardi, Challenges for Data Mining in Distributed Sensor Networks, *18th International Conference on Pattern Recognition (ICPR'06)*, p. 1000-1007 (2006).

[4] P. Sridhar, A.M. Madni, M. Jamshidi, Hierarchical Data Aggregation in Spatially Correlated Distributed Sensor Networks, *World Automation Congress (WAC '06)*, p.1-6 (2006).

[5] K. Romer, F. Mattern, The Design Space of Wireless Sensor Networks, *IEEE Wireless Communications*, **11(6)**, p. 54-61 (2004).

[6] S. Rajeev, A. Ananda, C. M. Choon, O. W. Tsang. Mobile, Wireless, and Sensor Networks - Technology, Applications, and Future Directions, John Wiley and Sons, 2006.

[7] C. Reed, M. Botts, J. Davidson, G. Percivall, OGC® Sensor Web Enablement: Overview and High Level Architecture, *IEEE Autotestcon, 2007*, p.372-380 (2007).

[8] A. Sheth, C. Henson, S. Sahoo, Semantic Sensor Web, *IEEE Internet Computing*, **12(4)**, p.78-83 (2008).

[9] M. Lewis, D. Cameron, S. Xie, B. Arpinar, ES3N: A Semantic Approach to Data Management in Sensor Networks, *Semantic Sensor Networks Workshop (SSN06)* (2006).

[10] P. B. Gibbons, B. Karp, Y. Ke, S. Nath, S. Seshan, Iris-Net: An Architecture for a Worldwide Sensor Web, *IEEE Pervasive Computing*, **2(4)**, p. 22–33 (2003).

[11] D. Moodley, I. Simonis, A New Architecture for the Sensor Web: The SWAP Framework, *Semantic Sensor Networks Workshop (SSN06)* (2006).

[12] N. Konstantinou, E. Solidakis, S. Zoi, A. Zafeiropoulos, P. Stathopoulos, N. Mitrou, Priamos: A Middleware Architecture for Real-Time Semantic Annotation of Context Features, *3rd IET International Conference on Intelligent Environments (IE'07)* (2007).

[13] M. Eid, R. Liscano, A. El Saddik, A Universal Ontology for Sensor Networks Data, *IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSA 2007)*, p. 59–62 (2007).

[14] D. Russomanno, C. Kothari, O. Thomas, Sensor Ontologies: from Shallow to Deep Models, *IEEE Computer Society*, p. 107–112 (2005).

[15] J. H. Kim, H. Kwon, D. H. Kim, H. Y. Kwak, S. J. Lee, Building a Service-Oriented Ontology for Wireless Sensor Networks, *Proceedings of the Seventh IEEE/ACIS International Conference on Computer and Information Science (ICIS 2008)*, p.649-654 (2008)

[16] M. Trifan, B. Ionescu, D. Ionescu, O. Prostean, G. Prostean, An Ontology based Approach to Intelligent Data Mining for Environmental Virtual Warehouses of Sensor Data, *IEEE Conference On Virtual Environments, Human-Computer Interfaces and Measurement Systems (VECIMS 2008)*, p.125-129 (2008).

[17] V. Huang, M. K. Javed, Semantic Sensor Information Description and Processing, *Second International Conference on Sensor Technologies and Applications, 2008 (SENSORCOMM '08)*, p.456-461 (2008).

[18] C. H. Jones, IEEE 1451.4 smart transducers template description language, http://standards.ieee.org/regauth/1451/IEEE_1451d4_TDL_Introduction_090104.pdf, accessed September 15, 2009.

[19] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, F. Silva, Directed Diffusion for Wireless Sensor Networking, *IEEE/ACM Transactions on Networking*, **11**, p. 2-16 (2003).

[20] S. Madden, M. Franklin, J. Hellerstein, W. Hong, TAG: A Tiny Aggregation Service for Ad-Hoc Sensor Networks, *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation* (2002).

[21] X. Li, Y.J. Kim, R. Govindan, W. Hong, Multi-dimensional Range Queries in Sensor Networks, *Proceedings of the 1st international conference on Embedded Networked Sensor Systems*, p. 63-75 (2003).

[22] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, F. Yu, Data-centric Storage in Sensornets with GHT, a Geographic Hash Table, *Mobile Networks and Applications*, **8(4)**, p. 427-442 (2003).

[23] S. Madden, M. Franklin, J. Hellerstein, W. Hong, TinyDB: An Acquisitional Query Processing System for Sensor Networks, *ACM Transactions on Database Systems (TODS)*, **30(1)**, p. 122-173 (2005).

[24] Y. Yao, J. Gehrke, The Cougar Approach to In-Network Query Processing in Sensor Networks, *SIGMOD Record*, **31(3)**, p. 9-18 (2002).

[25] Y. Yao, J. Gehrke, Query Processing for Sensor Networks, *First Biennial Conference on Innovative Data Systems Research (CIDR)* (2003).

[26] M. Sharaf, J. Beaver, A. Labrinidis, P. Chrysanthis, Balancing Energy Efficiency and Quality of Aggregate Data in Sensor Networks, *The VLDB Journal*, **13(4)**, p. 384-403 (2004).

[27] M. Cherniack, H. Balakrishnan, M. Balazinska, D. Carney, U. Çetintemel, Y. Xing, S. Zdonik, Scalable Distributed Stream Processing, *First Biennial Conference on Innovative Data Systems Research (CIDR)* (2003).

[28] S. Chandrasekaran, O. Cooper, A. Deshpande, M. Franklin, J. Hellerstein, W. Hong, S. Krishnamurthy, S. Madden, V. Raman, F. Reiss, M. Shah, TelegraphCQ: Continuous Dataflow Processing for an Uncertain World, *First Biennial Conference on Innovative Data Systems Research (CIDR)* (2003).

[29] D. Abadi, Y. Ahmad, M. Balazinska, U. Çetintemel, M. Cherniack, J.-H. Hwang, W. Lindner, A. Maskey, A. Rasin, E. Ryvkina, N. Tatbul, Y. Xing, S. Zdonik, The Design of the Borealis Stream Processing Engine, *Second Biennial Conference on Innovative Data Systems Research (CIDR)* (2005).

[30] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, P. Steggles, Towards a Better Understanding of Context and Context-Awareness, *Proceedings of the 1st international Symposium on Handheld and Ubiquitous Computing, LNCS*, **1707**, p. 304-307 (1999).

[31] G. Papamarkos, A. Poulovassilis, P. T. Wood, Event-Condition-Action Rule Languages for the Semantic Web, *Workshop on Semantic Web and Databases (SWDB 03)*, p. 309–327 (2003).

[32] S. Harris, N. Gibbins. 3store: Efficient Bulk RDF Storage, *Proceedings of the 1st International Workshop on Practical and Scalable Semantic Systems (PSSS'03)*, p. 1–15 (2003).

[33] O. Corby, R. Dieng-Kuntz, C. Faron-Zucker, Querying the Semantic Web with the Corese Search Engine, *Proceedings of the 15th European Conference on Artificial Intelligence (ECAI 2004)*, p. 705–709, (2004).

[34] J. Broekstra, A. Kampman, F. van Harmelen, Sesame: An Architecture for Storing and Querying RDF Data and Schema Information, *Towards the Semantic Web*, p. 71–89, John Wiley & Sons, Ltd, DOI: 10.1002/0470858060.ch5, (2003).

[35] A. Dey, Understanding and Using Context, *Journal of Ubiquitous Computing*, **5(1)**, p. 4-7 (2001).

[36] F. Baader, W. Nutt, Basic Description Logics, in the Description Logic Handbook, p. 47–100, Cambridge University Press (2002).

[37] J. Cardoso, The Semantic Web Vision: Where are We?, *IEEE Intelligent Systems*, **22(5)**, p. 84–88 (2007)

[38] F. Donini, M. Lenzerini, D. Nardi, A. Schaerf, Reasoning in Description Logics, in Gerhard Brewka, ed., Principles of Knowledge Representation, p. 191–236. CSLI Publications, 1996.

[39] B. Motik, U. Sattler, A Comparison of Reasoning Techniques for Querying Large Description Logic ABoxes, *Proceedings of the 13th International Conference on Logic for Programming Artificial Intelligence and Reasoning (LPAR'06)*, LNCS, **4246**, p. 227–241 (2006)

[40] E. Sirin, B. Parsia, B. Grau, A. Kalyanpur, Y. Katz, Pellet: A Practical OWL-DL Reasoner, *Journal of Web Semantics*, **5(2)**, p. 51–53 (2007)

[41] D. Tsarkov, I. Horrocks, FaCT++ Description Logic Reasoner: System Description, *Proceedings of the International Joint Conference on Automated Reasoning (IJCAR 2006)*, LNAI, **4130**, p. 292-297 (2006).

[42] J. P . Walters, Z. Liang, W. Shi, V. Chaudhary, Wireless Sensor Network Security: A Survey, in Security in Distributed, Grid, and Pervasive Computing, Auerbach Publications, CRC Press, 2006.

[43] L. Kazatzopoulos, C. Delakouridis, G. F. Marias, and P. Georgiadis, iHIDE: Hiding Sources of Information in WSNs, *Second International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing (SecPerU 2006)*, p. 41–48 (2006).

[44] O. Erling, I. Mikhailov, RDF Support in the Virtuoso DBMS, *Proceedings of the 1st Conference on Social Semantic Web (CSSW 2007)*, LNI, **113**, p. 59-68 (2007).

[45] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, DBpedia: A Nucleus for a Web of Open Data, *6th International Semantic Web Conference (ISWC 2007)*, p. 11-15 (2007).

[46] B. Warneke, M. Last, B. Liebowitz, K. S. J. Pister, Smart Dust: Communicating with a Cubic-Millimeter Computer, *Computer*, **34(1)**, p. 44-51 (2001).

[47] J. Yick, B. Mukherjee, D. Ghosal, Wireless Sensor Network Survey, *Computer Networks*, **52(12)**, p. 2292-2330 (2008).

[48] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless Sensor Networks: a Survey, *Computer Networks*, **38(4)**, p. 393-422 (2002).

[49] A. Zafeiropoulos, E. Solidakis, S. Zoi, N. Konstantinou, P. Papageorgiou, P. Stathopoulos and N. Mitrou: A lightweight approach for providing Location Based Content Retrieval, *In 18th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'07)*, Athens, Greece (2007)