

Document downloaded from:

<http://hdl.handle.net/10251/54652>

This paper must be cited as:

Giménez Salazar, P.; Molina Moreno, B.; Calvo-Gallego, J.; Esteve Domingo, M.; Palau Salvador, CE. (2014). I3WSN: Industrial Intelligent Wireless Sensor Networks for indoor environments. *Computers in Industry*. 65(1):187-199. doi:10.1016/j.compind.2013.09.002.



The final publication is available at

<http://dx.doi.org/10.1016/j.compind.2013.09.002>

Copyright Elsevier

Additional Information

I3WSN: Industrial Intelligent Wireless Sensor Networks for Indoor Environments

Pablo Giménez⁺, Benjamín Molina⁺, Jaime Calvo-Gallego^{*}, Carlos E. Palau¹⁺, Manuel Esteve⁺

⁺ Departamento de Comunicaciones, Universitat Politècnica de València 46022 Valencia, (Spain)
{pabgisa;benmomo;cpalau;mesteve}@upvnet.upv.es

^{*} Department of Computing and Automatics, E.P.S. de Zamora, University of Salamanca, 49022
Zamora, (Spain)
jaime.calvo@usal.es

ABSTRACT

Sensor Web Enablement (SWE) technologies have been successfully applied to a great variety of outdoor scenarios but, in practical terms, little effort has been applied for indoor environments, and even less in the field of industrial applications. This article presents an intelligent SWE application for an indoor and industrial scenario, with the aim of improving and increasing the levels of human safety. The base low-level architecture is built on top of wireless sensor networks (WSN) connected to a sensor observation service (SOS). Higher layers in the architecture include services that make real time decisions based on the collected data. Both simulation and experimental results are presented. The paper shows the viability of our approach in terms of performance, scalability, modularity and safety.

Keywords. - Indoor environments, industrial environments, industrial safety, sensor observation service (SOS), sensor web enablement (SWE), wireless sensor networks (WSN).

1. INTRODUCTION

Wireless Sensor Networks (WSNs) are a set of small, low cost and low energy devices that monitor an area of interest, capturing physical measurements (e.g. humidity, acceleration, pressure, etc.) and sending them to a sink node. The sink node connects the WSN with a powerful network node called Base Station (BS) where the sending data is stored and processed according to the scenario of application, which may range from agriculture, maritime, healthcare, industrial and even military applications.

Wireless Sensor Networks are one of the most important elements in the Internet of Things (IoT) paradigm, as they provide a virtual layer where any piece of sensing capable information about the physical world can be potentially accessed by any computational system. In this approach either the BS acts as gateway for the entire WSN or each sensor node may be directly accessed if it implements a TCP/IP stack. The former one refers to the typical scenario whereas the latter one is being developed and tested through standards such as 6LowPAN, defined by IETF, which allows transmitting IPv6 packets through computationally restricted networks [1].

Even if WSNs are becoming more and more popular they are still subject of huge research in various fields. Most of the research has been placed on reducing energy consumption [2] [3], developing efficient routing protocols [4] [5] and introducing or enhancing security issues [6] [7].

Considering the application scenarios, WSNs are typically deployed on outdoor environments, such as rivers, maritime coasts, gardens, large estates, surrounding areas (intrusion detection systems), or even battlefields (human tracking systems). For such outdoor scenarios there is often integration with Geographic Information System (GIS) applications. The GIS is useful to provide a visual layout of hundreds of sensors on a screen, even if such sensors do not change their location (they are typically identified by GPS coordinates). In the case of human or machine tracking systems, a GIS is really

¹ Corresponding Author

useful for monitoring target mobility and make decisions on their location. Unfortunately GIS applications are often limited to outdoor scenarios; they are not used for indoor scenarios due to several reasons:

- It is not possible to obtain a GPS location in indoor environments (buildings, factories, tunnels, etc.) as the GPS signal strength drastically reduces inside buildings.
- It is really difficult to accurately locate a target for indoor scenarios. The location is often estimated based on radio measurements which may vary significantly over time. This normally implies analyzing and modeling the radio channel per each site.
- It is expensive to deploy an indoor location system as it typically requires large equipment and processing power, depending on the size of the building/factory/tunnel and the required accuracy.

There are also other types of limitations when trying to deal with different sensors and thus with different WSNs regarding interoperability. Unfortunately, a normal WSN often focuses on a correct communication management scheme among all nodes, and all sensing data is stored in the BS inside a database in a particular way. Such particular way must be known by high-level applications if they want to retrieve sensing information. For a different WSN, the same high-level application must know how to access another sensing information database, which makes the application highly dependent on the WSN that is being used. In theory, the application only requires to know the sensed data to start processing on it, unaware of the WSN that is providing such data. For such situations, interoperability is offered in two main ways:

- Accessing directly each sensor node, via IPv6 (6LowPAN). In this case the application must issue a request per each sensor node it has to contact. This implies the usage of a node registry where all sensors from all available WSNs are stored.
- Accessing an intermediate registry that acts as ‘global base station’ for all WSNs with certain special features that make such usage particularly useful. This approach has been specified by the Open Geospatial Group (OGC) as a Sensor Observation Service (SOS) [8].

SOS is a standard specification included in the Sensor Web Enablement (SWE) framework [9]. SWE is a collection of open interoperability interfaces, metadata encodings and services for exploiting Web-connected sensors and sensor systems of all types. Developers and system integrators can use these specifications for building applications, platforms, and products involving Web-connected devices such as flood gauges, air pollution monitors, stress gauges on bridges, mobile heart monitors, webcams, etc. It is quite probable that for any current operational environment there is at least one useful SWE application that will enhance performance or even add new features. Web Sensors and SWE are in fact concepts in the context of GEOSS (The Global Earth Observation System of Systems) [10].

This paper describes the usage of intelligent wireless sensor networks in indoor industrial applications (I3WSN). The main goals addressed by the proposed system are:

- Intelligent: our system uses an interoperable scheme (SOS) where different WSNs can register and different applications can retrieve data to make decisions.
- Indoor: our system uses a GIS application integrated with an indoor location system that maps internal locations offsets to GPS coordinates.
- Industrial: our system has as target application scope an industrial manufacturing environment, where both machines and workers within the factory are tracked in order to assess risks in real time and avoid therefore any potential accident. Whereas the focus of industrial applications are normally the machines (availability, accuracy, security), the main focus in this paper is mainly the worker as crucial asset for the factory. The European Factories of the Future (FoF) concept [11] defines a framework where worker’s health and safety is a key aspect to improve competitiveness and productivity, and to the extent of our knowledge our work is the first application of SWE for the FoF environment.

The rest of the paper is structured as follows. The second section presents the motivation and related work. Section 3 provides the detailed description of the overall architecture and the individual components. Performance evaluation of the I3WSN proposal is presented in the fourth section and the paper finishes with the conclusions and future work.

2. MOTIVATION AND RELATED WORK

The main motivation of the work presented in this paper is the application of SWE technology to the FoF environment in order to increase levels of industrial safety, and specifically to its SOS component; location estimation and also semantics and intelligence. There is little research regarding this field where the operational environment refers to industry and critical operations. In [12] the problem of localizing and tracking mobile nodes acting in a fixed WSN is addressed, and the proposed approach results in being well suited for low-density setups and critical environments. Our paper goes beyond tracking mobile nodes as it also includes making real time decisions based on the worker's location in order to proactively assess potential risks and therefore improve worker's safety. The FASyS project [13], which encompasses the work done in this paper and more, has already tackled the specific issue of indoor positioning in industrial environments [14] [15]. Risk assessment is evaluated in [16] for collaborative robots in a human shared environment, following the relevant guidelines in the standards ISO 10218 [17] and ISO 13849 [18] to determine the requirements on the implementation of the risk reduction measures used. On the other hand, the work done in this paper is more focused on the standard ISO 31000. [19] This standard provides generic guidelines for the design, implementation and maintenance of risk management processes throughout an organization, but our paper will focus on the machinery industry group. Considering the worker as the nuclear entity of the risk assessment process, its health condition and health information is also of significant relevance to evaluate safety risks. [20]

Sensor web technologies (SWE) have been successfully applied to a large variety of outdoor scenarios. In [21] authors present an Internet based urban environment observation system that is able to monitor several environmental variables (temperature, humidity, illumination or air) in urban areas in real time. The environmental data is archived and later on retrieved through a SOS. Ocean Sensor Web (OceanSW) is described and prototyped in [22], whose main goal was to describe, organize, store and manage the ocean sensor data. The OceanSW system data is encoded and represented in XML, which is easy to exchange and manipulate. Furthermore the system uses an Ocean Sensor Observation Service. Another ocean-observing smart sensor web system described in [23] is composed of both mobile and fixed underwater ocean sensing assets as well as Earth Observing System satellite sensors that offer a large scale sensing feature. Authors in [24] present a coastal sensor web system that uses OGC SWE framework in order to achieve interoperability across various agencies and institutions that gather marine observations for oceanographic study, including a semantic middleware named COastal SEMantics MiddleWare (COSEMWare). This system goes beyond syntactic standardization of the metadata through open standards providing semantic enrichment in the form of ontologies. The usage of ontologies is not only applied to sensors, but also to services. In [25], a very specific application can be found oriented to interoperate power systems. Here sensor data from utilities (electrical systems) are published in Common Information Model (CIM) format and exposed via a SOS. This approach offers a standard method for discovering and accessing sensor data between utilities, which facilitates a rapid response to handle contingencies. Moreover the paper enhances CIM with semantics and uses Web Ontology Language (OWL) instead of Resource Description Framework (RDF) to overcome its limitations (e.g. cardinality).

Regarding indoor environments there are various different applications and systems developed to solve specific problems. Localization is one key technique in Wireless Sensor Networks as the popular GPS (Global Positioning System) does not work indoors. [26] To deal with this issue, there are several approaches for local positioning systems including Bluetooth, Wi-Fi, RFID, ultrasound, ultra-wideband (UWB) and WiMAX. A comparison of all of them can be found on [27]. For detailed accuracy (below 15 cm) only ultrasound and UWB systems are suitable, however the former one exhibits poor performance in the presence of ultrasonic noise [28] and thus is not recommended in industrial environments. Pulsed UWB communication seems an optimal candidate due to its ultrafine time resolution, robustness against multipath fading, low-cost architecture and low-power-consuming operation [29] [30]. Several low-power UWB-based indoor localization systems are described in [31] [32] [33], however it is still not a widely deployed technology.

Most location estimation research comes from the robotic field as it is a key issue. Classical methods for indoor location include: RSSI (Received Signal Strength); TOA (Time of Arrival); TDOA (Time Difference of Arrival); and AOA (Angle of Arrival). The general process of location-estimation

involves three main steps. The first one consists in retrieving distances or angles between nearby sensor nodes, either via RSSI; TOA; TDOA or AOA. The second step calculates the positions of location-unaware nodes considering the measurements obtained in the first stage. The third step is optional and implies one or more iterations to refine the location estimation [34]. Different proposals exist like the ones proposed in [35] or in [36]; and a detailed comparison between algorithms and techniques for indoor positioning can be found in [37] [38].

Intelligence in an ICT system refers to how data is treated and processed within the system to be used to make relevant and useful decisions. There are two generic approaches: make applications and services smarter or make data smarter. Semantics and metadata are a common way to enrich the available data and/or services in order to allow reasoning, as is done in COSEMWare [24]. A semantic SOS (semSOS) is presented in [39] where the domain of sensors and sensor observations is modeled in a suite of ontologies, adding semantic annotations to the sensor data, and using ontology models to reason over sensor observations. This semantically enabled SOS provides the ability to query high-level knowledge of the environment as well as low-level raw sensor data.

There are several approaches for adding semantics to sensor observations [40] [41] [42]. The concept of Semantic Sensor Web (SSW) is often used; within this concept sensor data is annotated with semantic metadata to increase interoperability, as well as to provide contextual information essential for situational knowledge. In particular, this involves annotating sensor data with spatial, temporal, and thematic semantic metadata. [43]

3. SYSTEM ARCHITECTURE

I3WSN system architecture is organized hierarchically, covering areas, zones and a global control center, as depicted in Figure 1. In the architecture, an area is considered to be a small location of interest where sensors have been deployed to monitor one or more features. A zone is a moderate location that aggregates several areas in order to monitor combined situations and risks, and may require the fusion of data from the covered areas. All sensor data collected in a zone is stored in a local SOS. A Local Control Center (LCC) exists in each zone and is in charge of processing the gathered data applying risk assessment algorithms that detect a potential problem or local risk for the worker and reacting in real time with reduced response time.

The aggregation of all data from all zones in the factory is performed at the Global Control Center (GCC), in order to detect global risks and make global decisions related with safety in the factory, and is composed of several blocks:

- HMI (Human Computer Interface): provides access to the whole system, allowing system configuration (locally and globally); setting up workers, sensors, areas, risks, etc. Besides configuration, administrators can also display real time information about what is happening in the factory (risk levels, alarms, location of workers, etc.)
- Event Processing: is in charge of: (i) monitoring all data that is being sensed and (ii) matching configuration conditions in order to detect potential risks. As the amount of information to process is huge, a Complex Event Processing (CEP) is included.
- Action Handler: launches and monitors the actions that have to be performed once an alert has been generated by the Event Processing module.

The GCC may also contact other systems and applications to gather additional information. The systems, applications or services may be internal or external depending on its availability in the time of deployment of I3WSN. On one hand, the internal ones refer to existing systems, applications or services that are already available in the factory prior to deploying the I3WSN system, e.g.: a factory has already a database with basic and personal information of every worker. The GCC may correlate information coming from sensors (I3WSN system) and workers (existing internal application) to make relevant decisions and infer potential risks. On the other hand, the external ones are systems, applications or services available on the Internet to optimize or enhance I3WSN performance. This is an extra optional feature that will not be targeted in this paper, as I3WSN will be described as an autonomous and independent system. However, the Internet may also serve as communication channel for sending alerts and notifications to remote workers and managers.

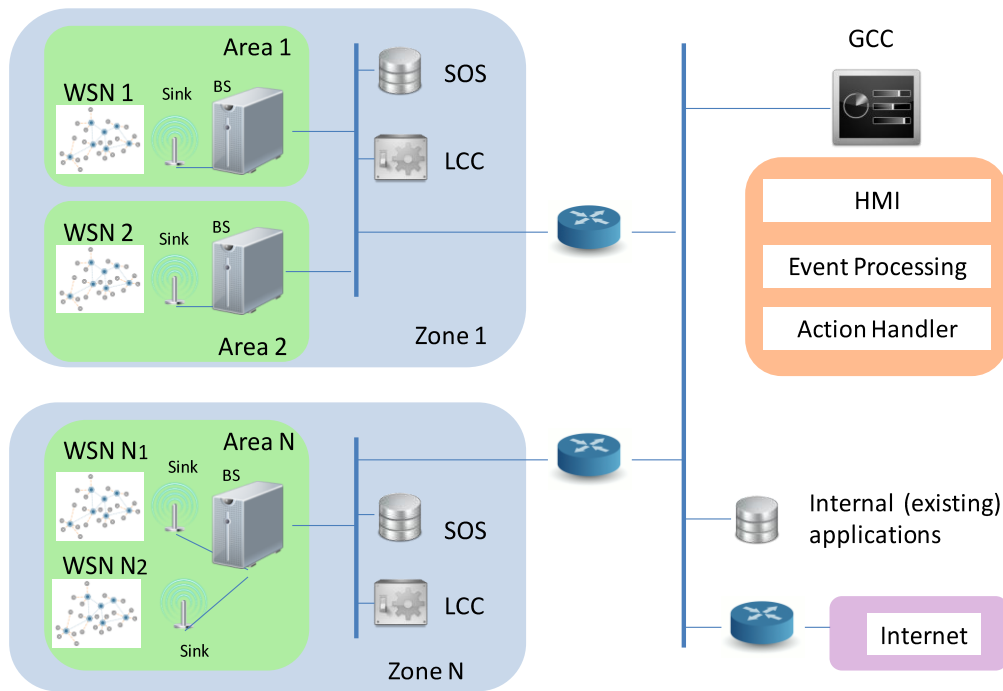


Figure 1. Global system architecture

Factory organization into zones and areas depends on its size and its activity, e.g. for a small factory an area and a zone may overlap and be identical; only one SOS would be required and both LCC and GCC may refer to the same control center. For a medium factory, several possibilities may arise: an area may refer to a specific physical area (production line, store, offices, etc) or to a specific logical area (in which a special WSN has been deployed; a special feature of interest is being monitored; etc). For a large factory consisting in various sites, a zone may refer to each separate site.

As can be observed in Figure 1, there are several entities in the architecture. The entities rely on communications to exchange information between them. The architecture is hierarchical also in terms of communications, and different network levels may be identified:

- Area level network (intra-area network): it corresponds to one or more WSNs consisting of several sensors monitoring one or more features. This is the lowest network communication level (sensor level) where sensors communicate and feed the system with data.
- Zone level network (inter-area network): this network connects several WSNs with a SOS and a Local Control Center (LCC). Each server (BS) from its correspondent area sends measurements to the SOS; therefore all sensor measurements within that area are accessible in the SOS through a standard interface (SensorML and O&M). Depending on the way the zone has been set up, this communication may occur through a wired or a wireless network.
- Control level network (inter-zone network): this network connects several zones together and allows the gathering of all necessary information in the GCC. Contrary to the LLC, the GCC must be aware of all zones as it contacts the SOS from each zone.

3.1. Wireless communications and WSNs

Communications in I3WSN as in other industrial applications involving sensors and sensor networks rely on wireless technology due to deployment facility. Different wireless technologies may coexist in a factory and especially within the proposed I3WSN in order to support from short range to medium or long range communications. COTS wireless technologies include Wi-Fi (IEEE 802.11) and Bluetooth (IEEE 802.15.1), however they are not especially suited for industrial environments and therefore its usage is limited. Compared to Bluetooth, IEEE 802.15.4 offers lower transmission rates but less power consumption; it also allows the interconnection of a relatively high number of

nodes building a mesh topology. Such feature offers an easy deployment of flexible and robust networks, able to self-adapt and self-configure when the operational environment changes.

There are various technologies based on IEEE 802.15.4 standard, such as ZigBee [44], WirelessHART [45] and ISA100.11a [46]. The latter two technologies have been designed to support industrial environments as they incorporate various robust and reliability mechanisms. On the other hand, though ZigBee was initially designed for home automation environments it was enhanced (ZigBee Pro or ZigBee 2007) to comply with industrial requirements. Basically ZigBee Pro keeps the physical and MAC layers from IEEE 802.15.4 and provides network and application layers with enhanced security features.

Though WirelessHART and ISA100.11a are more suitable for industrial applications, ZigBee is also a relatively good technological choice due to its wider availability, interoperability and lower cost. Thus, for practical reasons, ZigBee has been chosen as base wireless communication technology for deploying I3WSNs in the factory. Anyway, changing from one technology to another (as WirelessHART and ISA100.11a gain presence in the market) is relatively seamless for I3WSN system. An analysis of WSN deployment in industrial environments has already been performed in [14] and [15] and is out of the scope of the paper.

ZigBee provides short-range area communications, including router nodes, sensors, actuators and coordinator nodes that carry out the WSN management. The coordinator node is responsible for transmitting the sensed data to the LLC through a wireless backhaul that covers medium range technologies for communications within the factory, and long range technologies for the transfer of the aggregated data to the GCC. The adopted technology for medium-range zone communication level is IEEE802.11 (Wi-Fi), whereas the technology for long range control communication level is mainly targeted by IEEE802.16 (WiMAX), though 3.5G/LTE is envisioned for certain connectivity options. Figure 2 shows I3WSN heterogeneous communications architecture for industrial environments. The adopted approach facilitates a hierarchical, scalable and cost-efficient deployment.

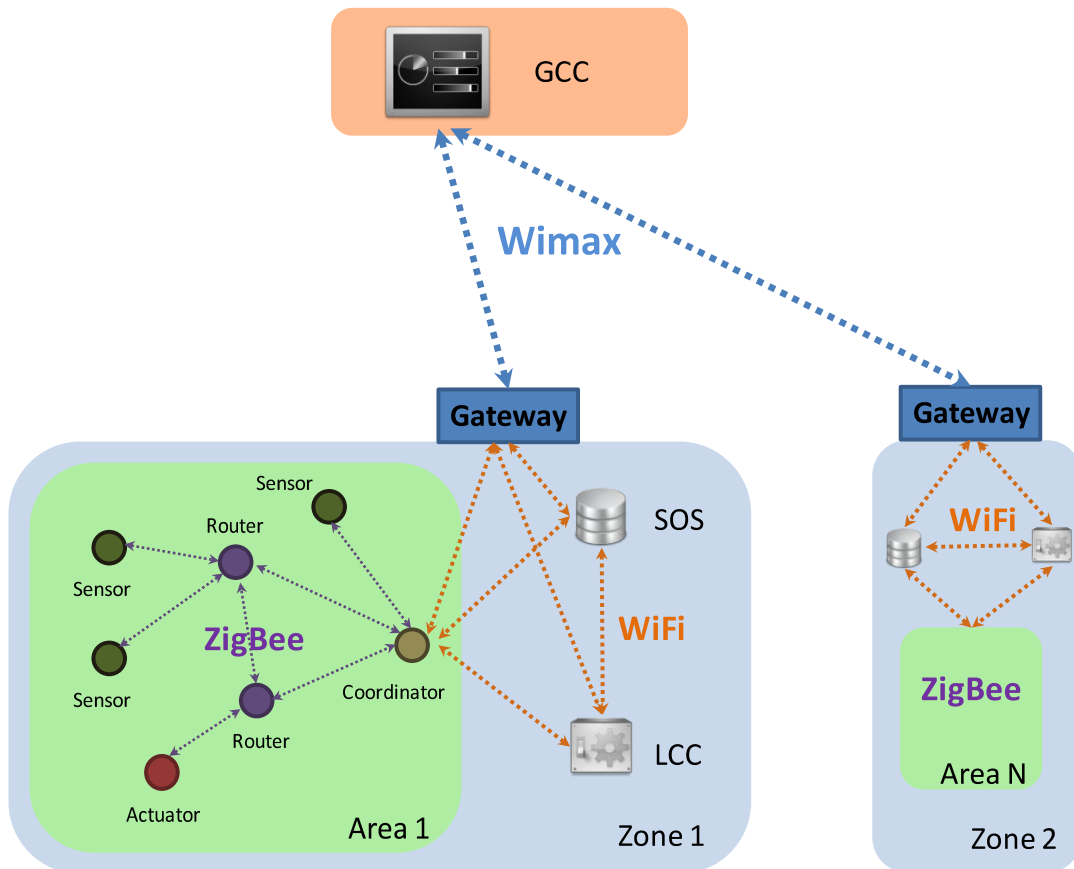


Figure 2. I3WSN heterogeneous communications architecture.

3.2. Sensor Observation Service and SWE

The Open Geospatial Consortium (OGC) created the Sensor Web Enablement (SWE) as a group of specifications covering sensor description, related data models and services that offer interoperability, access and control in a web-based environment. The SWE architecture is composed of two main models: the information model and the service model. The information model specifies the conceptual models and encodings whereas the service model specifies related services.

The conceptual models included in the information model specification refer to: (i) transducers (interfaces between real and digital world, which can be either sensors or actuators); (ii) processes (entities composed by functions and parameters that produce one or more outputs from one or more inputs); (iii) systems (group of georeferenced transducers that transform outputs from inputs according to a given methodology); and (iv) observations (the fact of observing a phenomenon including relevant information such as the value, the date and the location of the measurement).

The information model also includes a core suite of language and service interface specifications, such as:

- *Transducer Markup Language (TML)*: this language defines a model upon which sensor data can be sent (streamlined), archived, aggregated and analyzed efficiently and in a common way. TML was commonly used for sending live data from a sensor to a client [47] but is currently not being used (deprecated).
- *Sensor Model Language (SensorML)*: this language describes a common format for describing sensors and sensor systems, which facilitates sensor discovery as well as the analysis and processing of sensor data. Therefore sensors can be modeled as processes. [48]
- *Observation and Measurements (O&M)*: this specification provides a standard model for describing and exchanging observation results. An observation refers to an event happening at a certain time that generates a value from an observed phenomenon. Other measurements properties can also be included (e.g. location). [49][50]

The service model describes the SWE framework services and it includes: Sensor Alert Service (SAS) [51]; Sensor Planning Services (SPS) [52]; Web Notification Service (WNS) [53]; Catalog Service Web (CSW) [54]; and Sensor Observation Service (SOS) [55].

WSN has used SOS in the architecture. The main purpose of this service consists in providing access to sensor observations in a standard way. The SOS complies with the O&M specification for modeling sensor observations, and with the *SensorML* specification for modeling sensors and sensor systems. The SOS specification defines different operations on various profiles:

- The core profile provides the basic functionality of a SOS:
 - Identification of available sensors (*GetCapabilities*)
 - Access to sensor data (*GetObservation*)
 - Description of a sensor (*DescribeSensor*)
- The transactional profile provides the following operations:
 - Register a new sensor (*RegisterSensor*)
 - Insert a new measurement (*InsertObservation*)
- The enhanced profile specifies additional operations currently not included in any developed SOS on the market.

The basic flow diagram regarding SOS operations is depicted in Figure 3. First, the producer entity (application controlling a sensor or a WSN, such as the coordinator node or BS) registers the sensor in the SOS by means of a *SensorML* message where the main features are provided (initial position, mobile or not, observed properties or inputs sensed, etc.). Note that the producer registers two sensors, 1 and 2, which seems possible for a coordinator that handles and manages a WSN. On registration success, the SOS entity replies with a unique ID that identifies the sensor for subsequent operations. After registration, the coordinator can insert observations (retrieved by the sensors) in the SOS entity. Though this operation is asynchronous, typically WSNs are configured to report sensed data periodically.

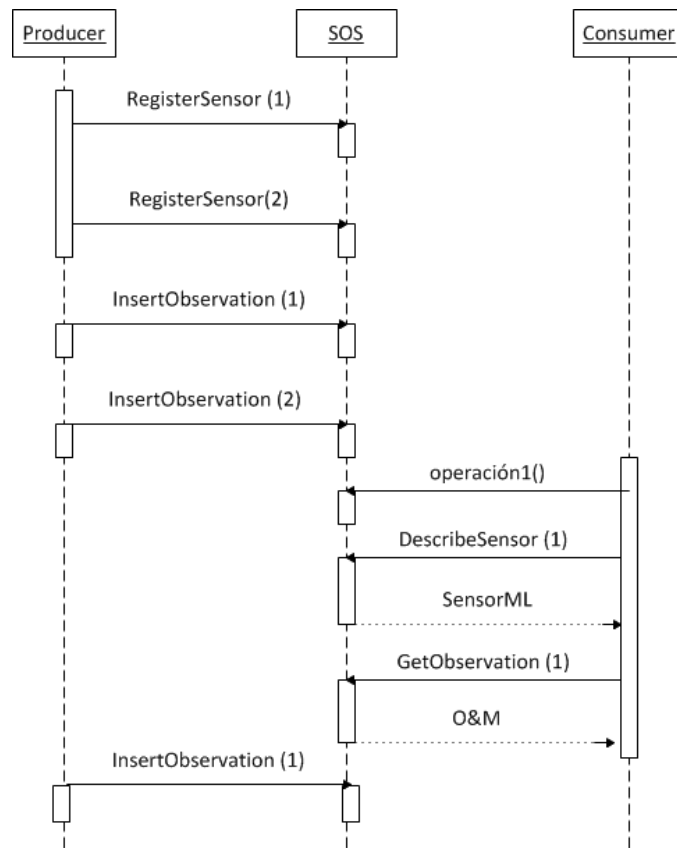


Figure 3. Basic sequence diagram for the SOS service

From another perspective, the consumer accesses the SOS entity in order to obtain information from the sensors. As the consumer is initially unaware of available sensors, it performs a *GetCapabilities* operation in order to retrieve a general description of available sensors. The *DescribeSensor* operation provides detailed information of a specific sensor and can be used by the producer to obtain all available parameters of an individual sensor. In fact, the description is a *SensorML* message similar to that inserted by the producer in the *RegisterSensor* operation. In order to obtain sensor measurements, the consumer contacts the SOS entity by means of a *GetObservation* operation. The SOS entity replies with an *O&M* message that describes the measurements taken by a specific sensor. Here again, the *O&M* object retrieved is similar to that provided by the sensor in the *InsertObservation* operation.

There are different developments of the SOS standard. However, they have several limitations that require adaptations to be performed in order to target all necessary requirements imposed by a particular project or deployment. The SOS server used in the I3WSN prototype has been developed by the 52north Sensor Web Community [56] and is currently the best open implementation available for research and production environments. The sensor sends basically two types of parameters to the SOS: location information (latitude, longitude) and observed properties (chemical concentrations, humidity, sound intensity, etc.) including altitude (that is not included in the original version and is needed for 3D location of the mobile sensors). Through a *RegisterSensor* operation, the sensor registers in the SOS describing its basic features, i.e. *identifier*, *status*, *mobile*, *sensorPosition* and *inputs and outputs*.

After the registering of a sensor, using the *DescribeSensor* operation the external applications will be able to retrieve from the SOS server the features of any registered sensor. The message flow exchange is depicted in Figure 4. First the producer (coordinator node) registers a sensor; later, as measurements arrive from the sensor, the coordinator node updates the sensor location through an *UpdateSensor* operation and an *InsertObservation* operation with the sensed data observed by the sensor. Sensors do not need to be real devices, and virtual sensors generated as computer processes can be deployed and provide observations from the environment of the factory. In every *RegisterSensor* operation a parameter indicating if a sensor is real or virtual is included.

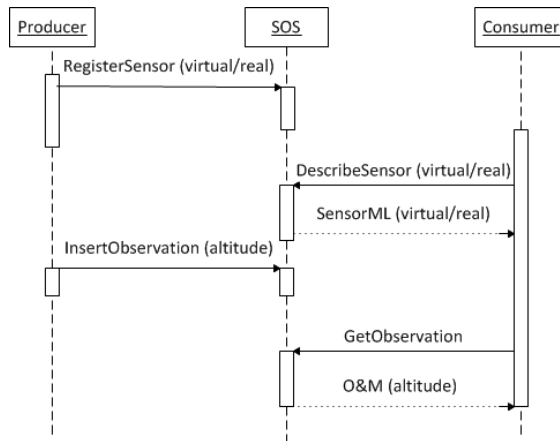


Figure 4. Enhancements to SOS implementation.

The use of virtual sensors turns out to be very useful to perform various types of tests. For example, one may start introducing many virtual sensors in order to evaluate the performance of the SOS and establish how many sensors a SOS may handle in proper working conditions. This will be treated in Section 4.1 and is a basic test to dimension correctly the capacity of a SOS and the number of required SOS servers within a factory. A more advanced test consists in evaluating the Event processing and Action handler processes for detecting special situations (potential risks or dangers) and react accordingly and in real time. The risk prevention manager can specify special dangerous situations within a factory, define the inputs given by sensed data, describe the alarms and finally establish the reaction process within a certain timeframe. By using virtual sensors, it is relatively easy to test and check whether the detecting algorithm is performing correctly and the Action handler is able to react within a specified timeframe; there is no need to invest on new WSN equipment until the test fulfills all requirements.

3.3. Local Control Center

The LLC is in charge of controlling local events that must be considered in a local scope, typically because they have real time requirements and decisions must be taken as soon as potential problems are detected in order to improve safety of the workers in an industrial environment. Therefore the LCC can be considered as a local decision point for special (real time) monitored events. In this case the decision does not scale up to the Global Control Center (GCC), as significant delay may convert the corresponding action in non-viable. However the information about the event and how it has been treated locally is sent to the GCC, as it stores all event information for making offline global analysis and detecting patterns that may lead to short term or long term incidents.

A basic example for an event controlled by a LLC is the potential collision between a worker and a machine (e.g. work lift or unmanned guided vehicle). If both the worker and the machine are moving within the factory, they may collide at a corner with low visibility, if they move along different paths that converge or intersect in such corner. Another possible situation is a static heavy machine with a worker moving around in the vicinity. If this heavy machine has moving metallic parts (e.g. machine arms) small metallic parts may become detached as result of the activity performed by the machine, then the worker is in potential danger if it is next to the working machine. This last example will be analyzed and evaluated in Section 4.

3.4. Global Control Center

The GCC is basically the kernel of the system, where mostly everything is configured and stored and important decisions are made as a result of evaluating every observation provided by the sensors deployed in the factory. The GCC is comprised of various modules. The detailed explanation of these modules is out of the scope of this paper and only the HMI will be described to some extent in order to highlight the main features of the system.

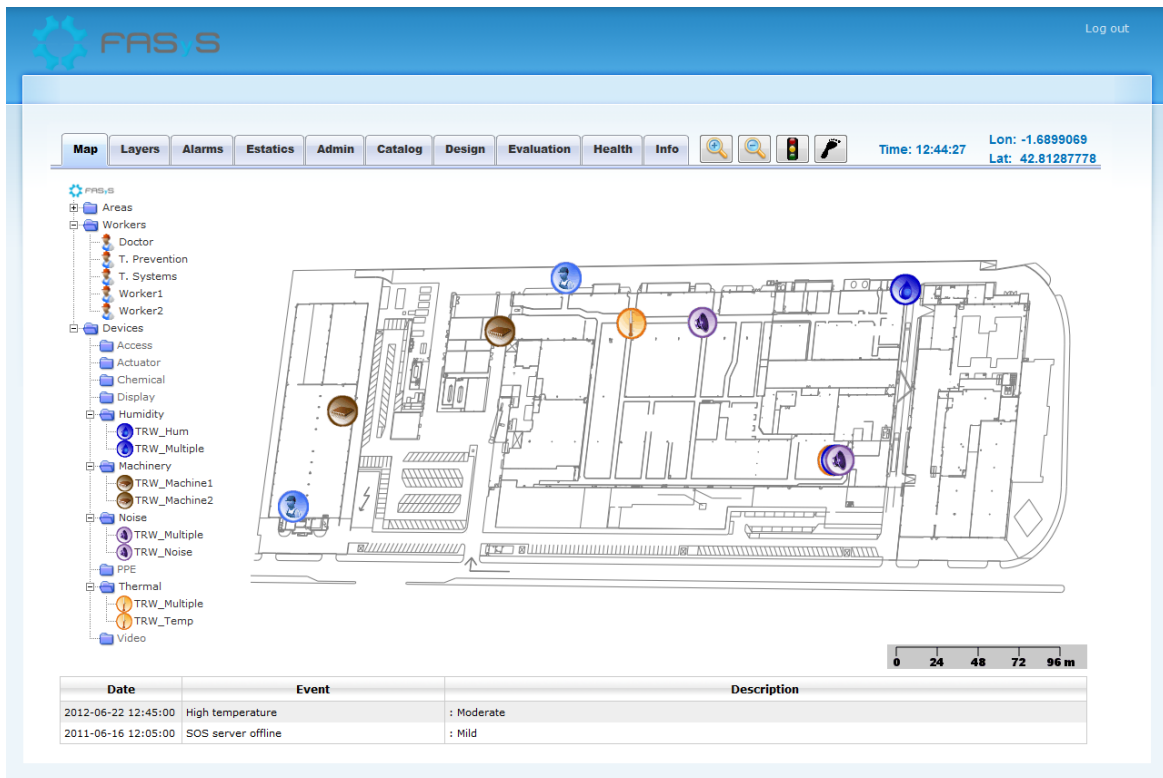


Figure 5. General monitoring screen.

3.4.1. Human Machine Interface

The HMI is the main interface that concentrates the administrative, monitoring and control functions of I3WSN in one single tool. The HMI application is web based and its structure is composed of three main areas, as shown in Figure 5:

- The top area lists a number of tabs (map, layers, alarms, statistics, admin, catalog, design, evaluation, health and info) corresponding to the main screens of operation.
- The central part is the part of displaying devices. It is the most relevant, visual and interactive part. The left zone displays a menu of the components, categorized into areas, workers and devices. The right zone is the visual georeferenced area (including a scale indicator) of the components at the factory. Clicking on any of the devices (sensors) can provide additional information, such as the physical property that is being monitored (e.g. temperature, pressure, humidity, etc.), or even the alarms that are associated to a particular sensor.
- The bottom part displays in real time the events and alarms that are taking place within the factory.

Typically a factory may deploy hundreds of sensors depending on the number of potential events and risks wanted to be monitored. In such circumstances it is more practical to monitor special areas within the factory. The HMI allows the administrator to specify certain areas as well as risk colors, similar to a traffic light, which facilitates the visual monitoring of special areas (see Figure 6). The risk level of an area is determined by the highest risk level of any of the devices within this area. Obviously, area risk levels change over time as (i) mobile devices change from area to area and (ii) alarms are being handled reducing the risk level within the corresponding area.

Another interesting feature of the HMI is its capability to display human traces in order to track the movement of a specific worker. In case of accidents where workers faint, it may be of interest to know whether those workers have been exposed to one or more chemical product that may have affected them according to their medical situation.

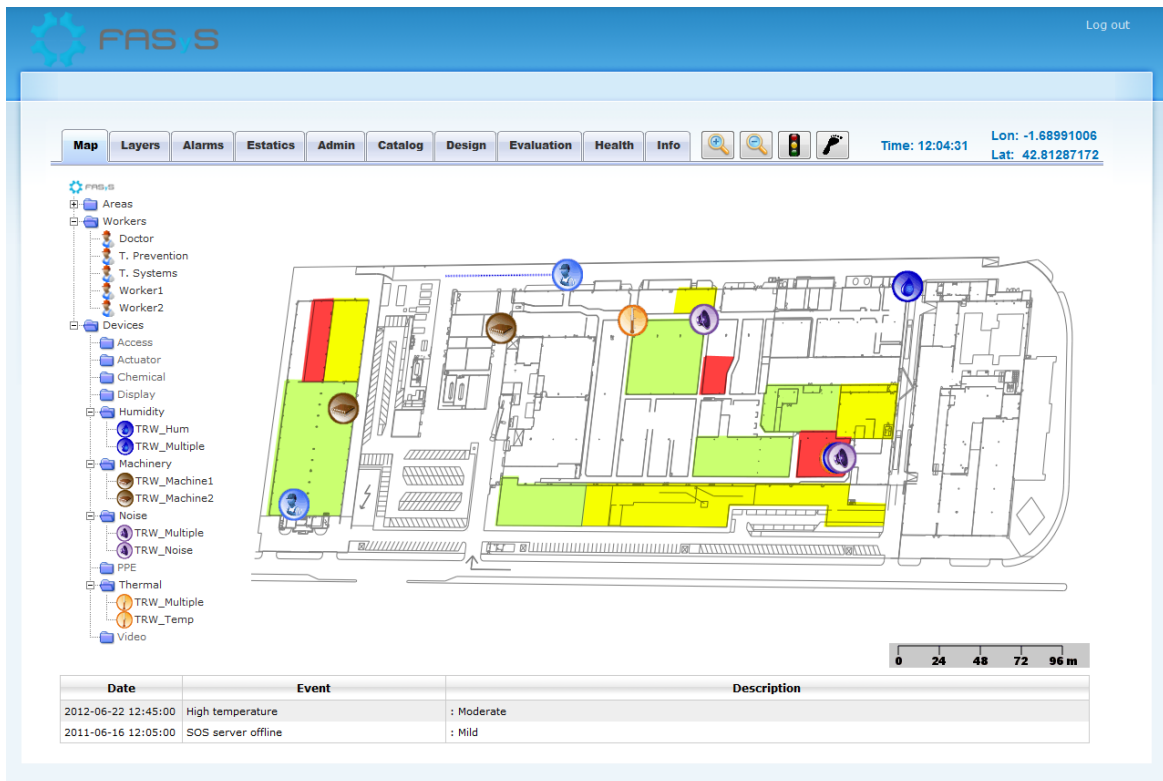


Figure 6. Risk Area Level management and traces.

3.4.2. Event processing

The Event Processing module is in charge of analyzing all inputs arriving from all factory areas in the form of observations. Factories are environments that potentially may generate large volumes of sensors and data, I3WSN includes a Complex Event Processing (CEP) tool. A CEP is able to process a great amount of incoming messages or events, regardless of whether they are historical or real-time in nature. A CEP can filter and analyze events in several ways, and perform actions with minimal latency when certain conditions of interest are triggered. The conditions can be simple or complex. A simple condition may consist of a sensor measuring a value above a certain threshold (e.g. the temperature in a particular area is too high). A complex condition may be a combination of the health status of a worker, several values of different sensors within a timeframe and the location of the worker (e.g. a worker suffering from asthma entering in a particular area in which he may be exposed to some chemical products for a certain period of time). The detailed description of the Event Processing module is out of the scope of this paper (as it is not directly focused on wireless sensors). [57]

3.4.3. Action Handler. Workflow

Once a risk has been detected by the Event Processing module, it is important to launch another module responsible for alleviating or avoiding such risk. Normally handling an event implies taking several actions, one after another or even several ones in parallel. The Action Handler in I3WSN has been developed using workflows, providing flexibility and simplicity in the configuration, as an administrator with basic skills can easily specify a response as the combination of several action items and let a workflow engine handle the execution and orchestration. The system captures the return value of each action item in order to ensure whether the global response for a particular event ended successfully or not. Note that most tracked events within the factory are worker oriented with a particular focus on its health and safety status, and therefore it is crucial to obtain a relatively high success ratio. This implies configuring several alternatives for the same response in order to enhance reliability. The detailed description of the Action Handler module is also out of the scope of this paper (not directly focused on wireless sensors) and can be considered as further work.

4. PERFORMANCE EVALUATION

The I3WSN system consists of several elements that, in general, allow a detailed tracking of working conditions in order to increase health and safety in a factory environment, with proactive actions to reduce or minimize risks. The evaluation of I3WSN focus on high level WSN aspects and the response that some of the components may provide. As the SOS has been targeted as an essential element of the system, it is relevant to evaluate this service and establish the amount of requests it can support in order to properly dimension the relationship between SOS servers and sensors deployed. Another aspect to evaluate is a special use case (collision detection) in order to test whether the current configuration is able to provide useful results in several working environments.

4.1. SOS performance

SWE-based systems (like I3WSN) are very dependent on the load that the SOS is able to support. In this first test we evaluated how many requests a SOS server can handle in order to limit the number of sensors that are assigned to such server. However it has to be considered that each deployment has its own individual and particular requirements, e.g. a small factory with a reduced number of deployed sensors may probably require only one SOS server; but a large factory may require several SOS servers covering different zones. However, this decision should be based on the amount of data generated by the observations of each sensor; the specific application and the structure of the underlying network. Therefore, it is important to determine the coverage or scope of a particular SOS server depending on its own resources (CPU, memory, network connection, storage capacity) and the number of sensors to be associated. It is difficult to compare sensors, as one sensor (e.g. temperature) may send periodical observations each minute whereas another sensor (e.g. chemical pollution) may send observations each second. There are also complex sensors that track and insert several physical *observedProperties* under the same offering at a time. From the point of view of the SOS server and its performance analysis the relevant figure is the number of *insertObservation* operations that can be performed, regardless of the number of sensors. Additional operations (besides *insertObservation*) provided by the SOS must be considered:

- *GetObservation*: the operation allows third party applications to query the SOS and obtain the sensed data, even with spatiotemporal filtering capabilities. For several reasons this operation is the most demanding one in terms of resources. Although requesting the last sensed value of a particular sensor does not require hardly any resource (unless the request is intensive and continuous), a client may ask for the whole history, which imposes (from the SOS perspective) getting all values, constructing an XML file and returning back it to the requesting client (third party application). Moreover, if spatial or temporal filtering operators are provided in the request, the SOS server has to perform additional operations consuming more resources. Prior to this operation, a third party application may perform two additional operations: *GetCapabilities* (for retrieving the available sensors in a SOS server in order to find a particular one) and *DescribeSensor* (for obtaining info describing this particular sensor).
- *UpdateSensor*: this operation allows mobile sensors to update their position within the factory. At least workers are considered mobile sensors, but some machinery (e.g. work lift) may also be considered mobile. Other sensors (e.g. temperature, humidity, chemical pollution, etc.) are typically deployed statically. Depending on the ratio mobile-fixed sensors, the *updateSensor* operation may or may not be taken under consideration. It is supposed that, in general, the movement of sensors (including workers) within the factory is not too fast and therefore the number of *updateSensor* operations is not intensive.
- *RegisterSensor*: the operation registers a particular sensor in the SOS server and provides a unique identifier to such sensor. This operation is not critical in terms of performance, if compared with *insertObservation* and *getObservation*. Note that sensors are registered beforehand and there is no special real time requirement in this operation.

For the tests we have used a Fujitsu Primergy TX100 S2 as SOS server, which consists of an Intel Xeon processor X3430 2.4 GHz, 4 GB RAM and 400GB HD. The implementation of the SOS corresponds to the latest available 52North SOS version [56] running on the Ubuntu Linux version

12.10. For the client part we have used Apache JMeter [57] as testing tool to generate the load and measure the performance. Five different load scenarios have been considered:

- Use Case 1: 20 users - 2500 requests
- Use Case 2: 50 users - 1000 requests
- Use Case 3: 100 users - 500 requests
- Use Case 4: 200 users - 250 requests
- Use Case 5: 500 users - 100 requests

The first case represents a reduced number of applications intensively interacting with the SOS server whereas the last case represents a large amount of applications slightly interacting with the SOS server. As the time interval for the evaluation is limited and equal for all cases, the request rate perceived by the SOS server is different during the simulation, even if the global number of requests is the same.

For the previous use cases we have considered three different request types generated by the clients:

- Request 1: *DescribeSensor*, representing the retrieval of positions inside the factory.
- Request 2: *GetObservation*, representing the retrieval of sensed data in the factory.
- Request 3: *GetCapabilities*, representing the retrieval of available sensors in the SOS.

For the combination of the request types and use cases we evaluated the following metrics: (i) the average, the median and the standard deviation of the delay, (ii) the delay, (iii) the throughput (number of processed requests), and (iv) the send and receive time.

The results of the load tests are provided in Figure 7, Figure 8 and Figure 9 where we present the average, the median and the standard deviation of the delay, as well as the throughput. As can be seen, case 5 is the most demanding one in terms of resources per time unit. Initially the SOS server is imploded by requests, but it is able to provide responses in acceptable times (100 ms in the worst case). For the *DescribeSensor* and *GetCapabilities* the SOS server is able to process up to 2000 requests per second. The *GetObservation* request is more CPU consuming, but the SOS is still able to provide up to 800 requests per second.

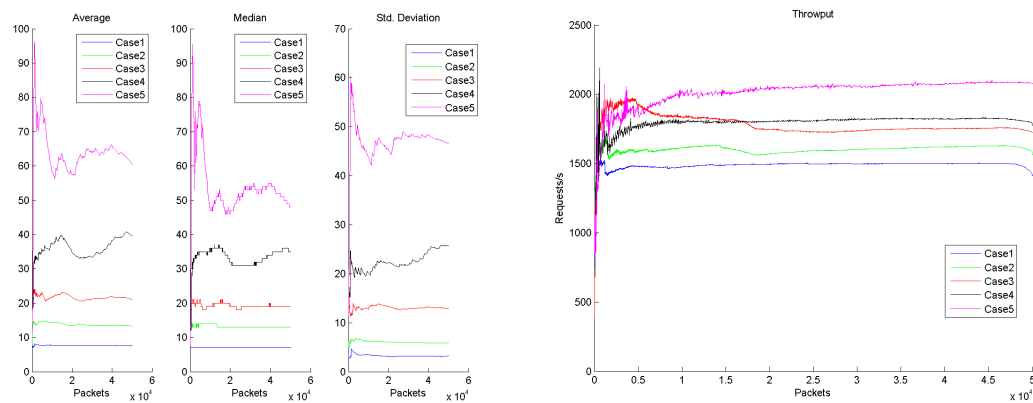


Figure 7. Simulation results for the *DescribeSensor* request.

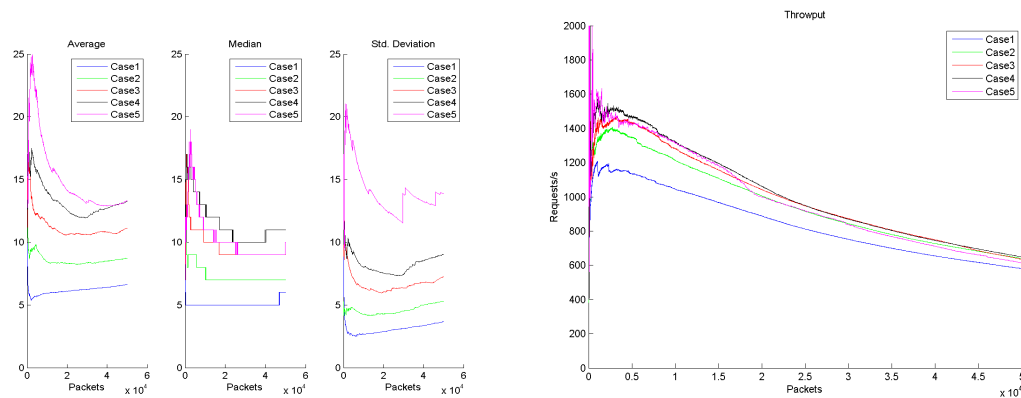


Figure 8. Simulation results for the *GetObservation* request.

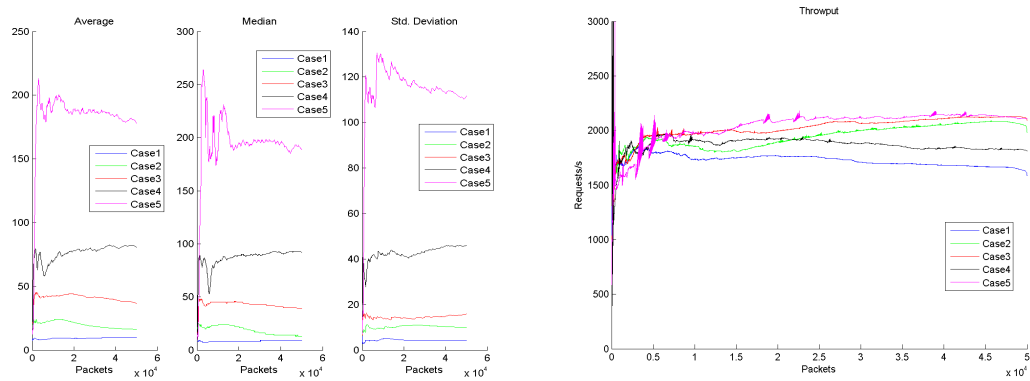


Figure 9. Simulation results for the *GetCapabilities* request.

4.2. Use case study

The second evaluation test of the I3WSN system consisted of an early (proactive) detection of a worker in the vicinity of a potential dangerous machine. In this case, when the worker approaches the working machine under a certain given distance, the latter must be stopped in order to avoid any type of accident. The scenario is depicted in Figure 10 and involves two actors: (i) the worker, which is represented by a position sensor indicating the location of the worker in the factory (mobile sensor) and (ii) the machine, which is also a sensor with a known location (fixed sensor). Around the machine there are two areas defined: (i) a risk area, where a worker may probably have an accident and (ii) a warning area, where the worker should be cautious not entering the risk area. The concept of warning area is in fact a predefined area to take a particular action. If the machine is potentially dangerous, the machine may be stopped. The warning area may also involve some alerting action.

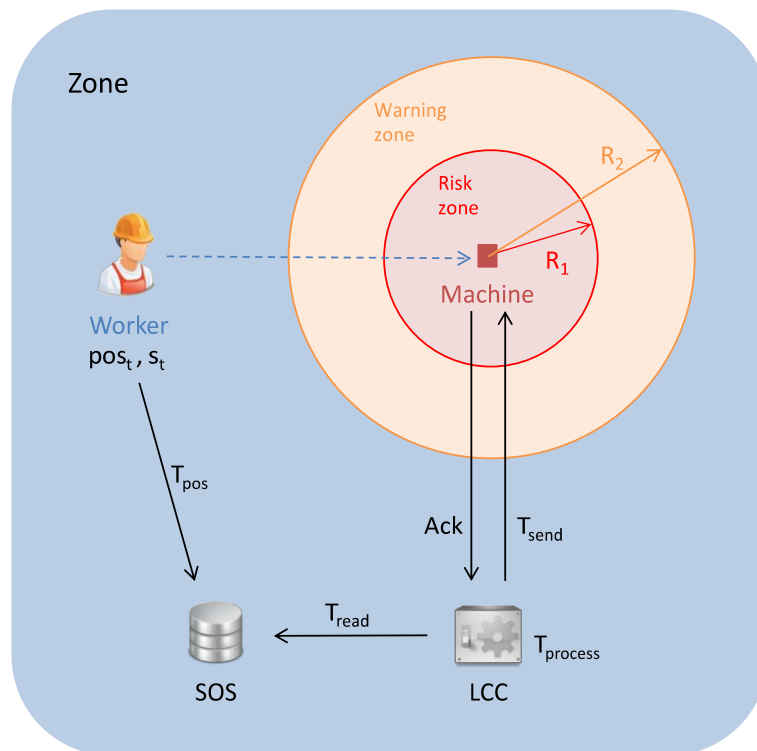


Figure 10. Use of case architecture

This use case is rather extensive in factories and is one of the key issues in workers safety in the FoF paradigm. The main objective is detecting when the worker approaches the machine under a R_2 distance (warning zone), and stop it before the worker reaches R_1 distance (risk zone). The LCC includes a small CEP responsible for monitoring this scenario and stopping the machine (in this case acting also as Action Handler for simplicity). The location information is obtained from the SOS, as the worker (i.e. considered as a mobile sensor) updates its location by means of periodical *UpdateSensor* operations against the SOS. The following parameters are relevant for evaluating the scenario (see Figure 10):

- The worker has a pos_i position and is moving at a s_i speed towards the machine.
- The worker position sensor sends its location to the SOS every T_{pos} (*UpdateSensor*).
- The LCC retrieves the worker's position every T_{read} (*DescribeSensor*) and calculates whether the worker is in the warning zone (under R_2 radius) or in the risk zone (under R_1 radius) in a time of $T_{process}$.
- If the worker is in the warning area, the LCC orders to stop the machine in a time of T_{send} and expects to receive the acknowledgment (ACK).
- By the time the LCC receives confirmation from the machine, the worker should not have entered the risk zone.

For the proposed scenario we have taken the following values for the parameters:

- Warning zone $R_2 = 3$ m.
- Risk zone $R_1 = 2$ m.
- The LCC reads information from the SOS each $T_{read} = 1000$ ms.
- The worker inserts its position into the SOS each $T_{pos} = 1000$ ms.

The scenario equipment was composed by the same SOS server used in section 4.1 and for the movement of the worker a sensor simulator, described in [59]. This simulator allows easily reproducing various movement types (e.g. random, linear, etc.) and inserting the values in the SOS for a configurable refresh interval (in this scenario each second).

The worst case in a linear movement is the worker moving directly to the machine (see path 1 in Figure 11), as the distance to traverse from R_2 to R_1 is directly radial (minimal) and therefore the LCC has less time to detect the event and react accordingly, compared to another non radial movement (see path 2 in Figure 11).

Various simulations have been carried out for different worker speeds and are reflected in Table 1. As can be observed there are various speeds of the worker where the LCC is unable to react in time and the worker enters part of the risk zone while the machine is already working. The reaction time varies from 110 to 120 ms, which is really small. As the worker's speed diminishes, the LCC is able to react in time, and for a speed below 2.6 m/s (see experiments 8 and 9) the machine is stopped before the worker enters the risk zone. Note that such speed (over 45 km/h) is by far greater than the speed of a worker or even a work lift in a factory, so it seems that the use case can be successfully deployed for these configuration parameters.

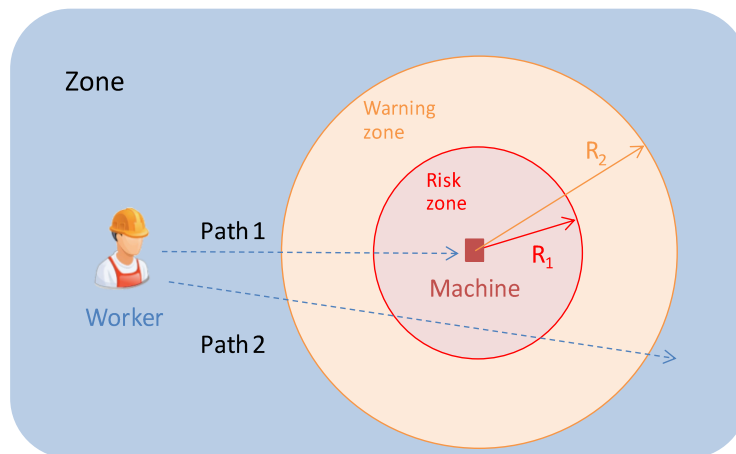


Figure 11. Different paths invading the warning and risk zone.

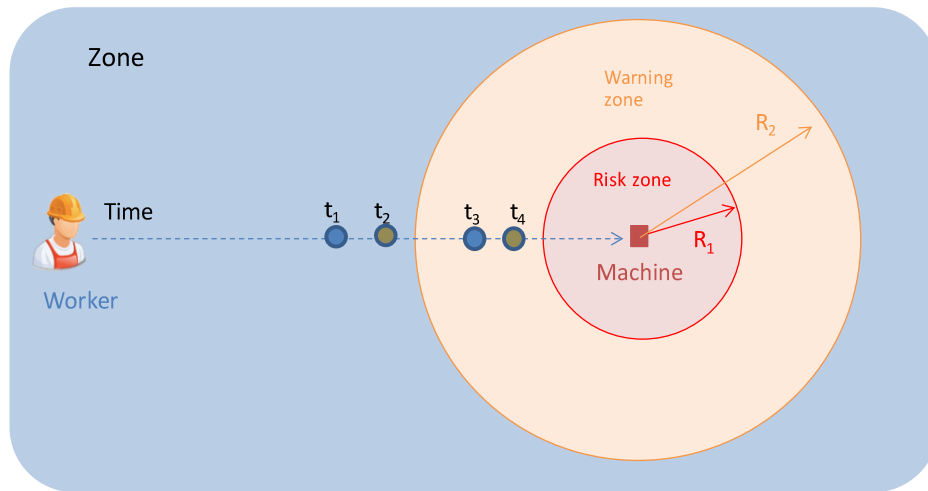


Figure 12. *UpdateSensor* and *DescribeSensor* operations.

ID	Warning zone detected at	Stopped after	Detected in the Risk Area	Speed
1	0.42452922 m (2.5754707 m entered)	110.0 ms	1.5754708 m inside	4.4896083 m/s
2	0.4250855 m (2.5749145 m entered)	120.0 ms	1.5749145 m inside	4.112429 m/s
3	1.2723997 m (1.7276003 m entered)	120.0 ms	0.72760034 m inside	3.7696009 m/s
4	1.2731794 m (1.7268206 m entered)	120.0 ms	0.7268206 m inside	3.5279293 m/s
5	1.2723997 (1.7276003 m entered)	110.0 ms	0.72760034 m inside	3.2846792 m/s
6	2.5445764 m (0.4554236 m entered)	110.0 ms	0.5445764 m inside	3.0078862 m/s
7	1.2723997 m (1.7276003 m entered)	100.0 ms	0.72760034 m inside	2.933788 m/s
8	2.751414 m (0.24858594 m entered)	120.0 ms	0.75141406 m outside	2.6957433 m/s
9	2.527801 m (0.47219896 m entered)	110.0 ms	0.52780104 m outside	2.6902618 m/s

Table 1. Use case simulation results

Sensors and LLC are not synchronized; although they share a common time base *updateSensor* and *DescribeSensor* operations take place at different times. The situation is depicted in Figure 12. The worker, as it approaches the machine, updates its position at t_1 and t_3 . However, the LCC requests the worker's position at t_2 and t_4 . Note here that the worker is not detected at the very moment it enters the warning zone, but some (little) time after (in the worst case nearly one second after, T_{read}). This could even lead to the case that, when the worker is detected in the warning zone, it is already in the risk zone. This happens when the worker moves terribly fast (see experiments 1 and 2).

5. CONCLUSIONS

Sensor web enablement (SWE) is a key modern technology that can be successfully applied to indoor environments. This paper presents a novel approach for using SWE technologies in industrial environments in order to monitor worker behavior and assure both their safety and health, in line with the FoF paradigm. In order to achieve such goal various heterogeneous wireless sensors are deployed within the factory, even those carried by workers in form of a Personal Area Network (PAN). The information sensed is sent to a Sensor Observation Service (SOS) server where it is stored and can be later accessed by applications in an interoperable way, as standard web interfaces have been specified for such purposes. This is a fundamental aspect for operation success as normally Wireless Sensor

Networks (WSNs) have their own communication interface and rarely interoperate, which represents a serious drawback for a scalable and modular deployment of sensor systems in a factory.

The incorporation of a SOS server allows a centralized repository of sensed information from which further analysis can be performed, events can be detected beforehand and proactive actions can be applied in order to minimize the risks for the worker in the factory. The system architecture has been designed in a modular and scalable way, so that it can be adapted to any factory size. A graphical user interface has been developed in order to easily monitor all actions taken place within the factory, even with indoor location systems. As the interface is web based, it allows easy integration in any user device (desktop, tablet and mobiles), which enhances the interaction and communication towards the worker through their mobile phone.

Several simulations have been presented in order to test the viability of the system. First, a load test to a SOS server has been performed in order to check whether such server is able to handle a potential large amount of requests. Note that SOS servers are typically used in outdoor environments with a read/write rate of minutes, hours or even days. This is not the case for industrial environments where data may be sensed in seconds for security or health purposes. In our experiments, the SOS server has shown a good performance. Additionally a real time use case scenario (collision detection) has been analyzed in order to check if real time prevention systems can be applied successfully. The results shown that workers may walk (run) at 45 km/h and collisions can be anticipated (and avoided).

Further work is oriented to test the performance of the SOS in a multiactor scenario, where multiple sensors and applications perform different operations to the SOS. Besides, the Event Handler should be further studied for more complex scenarios where multiple conditions may be monitored depending on the risk to be prevented.

ACKNOWLEDGMENTS

This work has been partially funded by the Spanish Ministry of Industry under the project FASyS (Absolutely Safe and Healthy Factory) grant number CENIT 2009-1034.

REFERENCES

- [1] G. Montenegro, N. Kushalnagar, J. Hui and D. Culler, "RFC 4944: Transmission of IPv6 Packets over IEEE 802.15.4 Networks", Internet Engineering Task Force, 2007.
- [2] V. Michopoulos, L. Guan, G. Oikonomou and I. Phillips, "DCCC6: Duty Cycle-aware Congestion Control for 6LoWPAN networks", in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 2012, pp 278-283.
- [3] C. Konstantopoulos, G. Pantziou, D. Gavalas, A. Mpitziopoulos and B. Mamalis, "A Rendezvous-Based Approach Enabling Energy-Efficient Sensory Data Collection with Mobile Sinks", *IEEE Transactions on Parallel and Distributed Systems*, 23 (5), 2012, pp. 809-817.
- [4] A.F. Liu, M. Ma, Z.G. Chen and W. Gui, "Energy-Hole Avoidance Routing Algorithm for WSN", in *Proceedings of the Fourth International Conference on Natural Computation (ICNC'08)*, 2008, pp. 76-80.
- [5] W. Zhang, Y. Wang and Y. Ma, "Research of WSN Routing Algorithm based on the Ant Algorithm", in *Proceedings of the 9th International Conference on Electronic Measurement and Instruments (ICEMI'09)*, 2009, pp. 422-426.
- [6] S. Ji, Q. Pei, Y. Zeng, C. Yang and S. Bu, "An Automated Black-Box Testing Approach for WSN Security Protocols", in *Proceedings of the 7th International Conference on Computational Intelligence and Security*, 2011, pp. 693-697.
- [7] W. Guo, Y. Hua and H. Ma, "Improving the Security of Time Synchronization in WSN", in *Proceedings of the IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, 2008, pp. 181-185.
- [8] __, "Sensor Observation Service (SOS)", Open Geospatial Consortium (OGC), available at <http://www.opengeospatial.org/standards/sos>, last visited May 2013.
- [9] __, "The OGC Sensor Web Enablement (SWE)", Open Geospatial Consortium (OGC), available at <http://www.opengeospatial.org/ogc/markets-technologies/swe/>, last visited May 2013.

- [10] E.J. Christian, "GEOSS Architecture Principles and the GEOSS Clearinghouse", *IEEE Systems Journal*, 2 (3), 2008, pp. 333-337.
- [11] __, "Research & Innovation Industrial Technologies, Public Private Partnerships in Research, Factories of the Future", available at: http://ec.europa.eu/research/industrial_technologies/factories-of-the-future_en.html, last visited May 2013.
- [12] A. Cenedese, G. Ortolan, and M. Bertinato, "Low-Density Wireless Sensor Networks for Localization and Tracking in Critical Environments", *IEEE Transactions on Vehicular Technology*, 59 (6), 2010, pp. 2951-2962.
- [13] Absolutely Safe and Healthy Factory (FASyS) project, available at <http://www.fasys.es/en/>, last visited May 2013.
- [14] M. Sepulcre, J. A. Palazón, J. Gozalvez, and J. Orozco, "Wireless Connectivity for Mobile Sensing Applications in Industrial Environments", in *Proceedings of the 6th IEEE International Symposium on Industrial Embedded Systems (SIES'11)*, 2011, pp. 111-114.
- [15] J. A. Palazón, M. Sepulcre, J. Gozalvez, J. Orozco, and O. López, "Heterogeneous Wireless Connectivity for Fixed and Mobile Sensing Applications in Industrial Environments", in *Proceedings of the 16th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'11)*, 2011, pp. 1-8.
- [16] B. Matthias, S. Kock, H. Jerregard, M. Kallman, I. Lundberg, and R. Mellander, "Safety of collaborative industrial robots: Certification possibilities for a collaborative assembly robot concept", *IEEE International Symposium on Assembly and Manufacturing (ISAM'11)*, 2011, pp. 1-6.
- [17] __, "ISO 10218-1:2006, Robots for industrial environments – Safety requirements – Part 1: Robot", *International Standard Organization*, 2006.
- [18] __, "13849-1:2006, Safety of machinery – Safety-related parts of control systems – Part 1: General principles for design", *International Standard Organization*, 2006.
- [19] __, "31000:2009, Risk management - Principles and guidelines", *International Standard Organization*, 2009.
- [20] G. Zülch and T. Grieger, "Modelling of Occupational Health and Safety Aspects in the Digital Factory", *Computers in Industry*, 56, 4, 2005, pp. 384-392.
- [21] X. Song, C. Wang, M. Kagawa and V. Raghavan, "Real-time Monitoring Portal for Urban Environment Using Sensor Web Technology", in *Proceedings of the 18th International Conference on Geo-informatics*, 2010.
- [22] Y. Jiang, J. Li, and Z. Guo, "Design and implementation of a Prototype System of Ocean Sensor Web", in *Proceedings of the IET International Conference on Wireless Sensor Network (IET-WSN'10)*, 2010, pp. 21-26.
- [23] B. M. Howe, Y. Chao, P. Arabshahi, S. Roy, T. McGinnis, and A. Gray, "A Smart Sensor Web for Ocean Observation: Fixed and Mobile Platforms, Integrated Acoustics, Satellites and Predictive Modeling", *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 3 (4), 2010, pp. 507-521.
- [24] S. S. Durbha, R. L. King, S. K. Amanchi, S. Bheemireddy, and N. H. Younan, "Standards-Based Middleware and Tools for Coastal Sensor Web Applications", *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 3 (4) 2010, pp. 451-466.
- [25] S.S. Durbha, R.L. King, and N.H. Younan, "Sensor web for interoperability in power systems", in *Proceedings of the North American Power Symposium (NAPS'09)*, 2009.
- [26] M. Gholami, N. Cai and R. W. Brennan, "Evaluating Alternative Approaches to Mobile Object Localization in Wireless Sensor Networks with Passive Architecture", *Computers in Industry*, 63, 9, 2012, pp. 941-947.
- [27] C. Benavente-Peces, V. Moracho-Oliva, A. Dominguez-Garcia, and M. Lugalde-Rodriguez, "Global system for location and guidance of disabled people: Indoor and outdoor technologies integration", in *Proceedings of the 5th International Network and Services Conference*, 2009, pp. 370-375.
- [28] A.M. Hazas, and A. Hopper, "Broadband ultrasonic location systems for improved indoor positioning", *IEEE Transactions on Mobile Computing*, 5 (5), 2006, pp. 536-547.

- [29] D. Porcino and W. Hirt, "Ultra-wideband radio technology: potential and challenges ahead", *IEEE Communications*, 41 (7), 2003, pp. 66-74.
- [30] S. Gezici et al., "Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks", *IEEE Signal Processing*, 22 (4), 2005, pp. 70-84.
- [31] K. Yu, and I. Oppermann, "Performance of UWB position estimation based on time-of-arrival measurements", in *Proceedings of the IEEE Conference on Ultra Wideband Systems and Technologies (UWBST'04)*, 2004, pp. 312-316.
- [32] G. Zhang, and S. V. Rao, "Position localization with impulse ultrawide band", in *Proceedings of the IEEE/ACES Conference on Wireless Communications Applied Computational Electromagnetics*, 2005, pp. 17-22.
- [33] Z. Li, W. Dehaene and G. Gielen, "A 3-tier UWB-based indoor localization system for ultra-low-power sensor networks", *IEEE Transactions on Wireless Communications*, 8 (6), 2009, pp. 2813-2818.
- [34] J. Albowicz, A. Chen, and L. Zhang, "Recursive Position Estimation in Sensor Networks", in *Proceedings of the IEEE International Conference on Network Protocols*, 2001, pp. 35-41.
- [35] S. Lee, B. Kim, H. Kim; R. Ha, and H. Cha, "Inertial Sensor-Based Indoor Pedestrian Localization with Minimum 802.15.4a Configuration", *IEEE Transactions on Industrial Informatics*, 7 (3), 2011, pp. 455-466.
- [36] L. Cheng, C.D. Wu and Y.Z. Zhang, "Indoor robot localization based on wireless sensor networks", *IEEE Transactions on Consumer Electronics*, 57 (3), 2011, pp. 1099-1104.
- [37] Y. Gu, A. Lo, and I. Niemegeers, "A survey of indoor positioning systems for wireless personal networks", *IEEE Communications Surveys & Tutorials*, 11 (1), 2009, pp. 13-32.
- [38] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of Wireless Indoor Positioning Techniques and Systems", *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37 (6), 2007, pp. 1067-1080.
- [39] C.A. Henson, J.K. Pschorr, A.P. Sheth and K. Thirunarayan, "SemSOS: Semantic Sensor Observation Service", in *Proceedings of the International Symposium on Collaborative Technologies and Systems (CTS '09)*, 2009, pp. 44-53.
- [40] D. Jeong, H. Jeong and Y.S. Jeong, "SS-RBAC: Secure Query Processing Model for Semantic Sensor Networks", in *Proceedings of the 2nd International Conference on Future Generation Communication and Networking (FGCN'08)*, 2008, pp. 352-355.
- [41] K. Thirunarayan, C.A. Henson, and A.P. Sheth, "Situation awareness via abductive reasoning from Semantic Sensor data: A preliminary report", in *Proceedings of the International Symposium on Collaborative Technologies and Systems (CTS '09)*, 2009, pp. 111-118.
- [42] V. Huang, and M.K. Javed, "Semantic Sensor Information Description and Processing", In *Proceedings of the 2nd International Conference on Sensor Technologies and Applications (SENSORCOMM'08)*, 2008, pp. 456-461.
- [43] A. Sheth, C. Henson, and S.S. Sahoo, "Semantic Sensor Web", *IEEE Internet Computing*, 12 (4), 2008, pp. 78-83.
- [44] J. Li, X. Zhu, N. Tang and J. Sui, "Study on ZigBee network architecture and routing algorithm", in *Proceedings of the 2nd International Conference on Signal Processing Systems (ICSPS'10)*, 2010, pp. 389-393.
- [45] J. Song, S. Han, A.K. Mok, D. Chen, M. Lucas, and M. Nixon, "WirelessHART: Applying Wireless Technology in Real-Time Industrial Process Control", in *Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'08)*, 2008, pp. 377-386.
- [46] ___, "Wireless Systems for Industrial Automation: Process Control and Related Applications, Standard 100.11a, Draft 2a", *International Society of Automation (ISA)*, 2009.
- [47] S. Havens, "OGC Implementation Specification 06-010r4: OpenGIS Transducer Markup Language", *Open Geospatial Consortium*, 2006.
- [48] M. Botts, "OGC Implementation Specification 07-000: OpenGIS Sensor Model Language (SensorML)", *Open Geospatial Consortium*, 2007.
- [49] S. Cox, "OGC Implementation Specification 07-022r1: Observations and Measurements—Part 1—Observation schema", *Open Geospatial Consortium*, 2007.

- [50] S. Cox, “OGC Implementation Specification 07-022r3: Observations and Measurements—Part 2—Sampling Features”; *Open Geospatial Consortium*, 2007.
- [51] I. Simonis, “OGC Best Practices 06-028r3: OGC Sensor Alert Service Candidate Implementation Specification”, *Open Geospatial Consortium*, 2006.
- [52] I. Simonis, “OGC Implementation Specification 07-014r3: OpenGIS Sensor Planning Service”, *Open Geospatial Consortium*, 2007.
- [53] I. Simonis, and J. Echterhoff, “OGC Best Practices 06-095r1: OpenGIS Web Notification Service Implementation Specification” *Open Geospatial Consortium*, 2007.
- [54] D. Nebert, A. Whiteside, and P. Vretanos, “OGC Implementation Specification 07-006r1: OpenGIS Catalogue Services Specification”, *Open Geospatial Consortium*, 2007.
- [55] A. Na, M. Priest, “OGC Implementation Specification 06-009r6: OpenGIS Sensor Observation Service (SOS)”, *Open Geospatial Consortium*, 2007.
- [56] 52 north Sensor Web Community, available at <http://52north.org/communities/sensorweb/>, last visited May 2013.
- [57] W. Fengjuan, Z. Xiaoming, W. Yongheng and C. Kening, “The Research on Complex Event Processing Method of Internet of Things”, in Proceedings of the 5th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), 2013, pp.1219-1222.
- [58] Apache JMeter, available at <http://jmeter.apache.org/>, last visited May 2013.
- [59] P. Gimenez, B. Molina, C.E. Palau and M. Esteve, “Semantic Sensor Web Simulator for Factory Automation”, In Proceedings of the 27th Symposium Nacional de la Unión Científica Internacional de Radio (URSI 2012), (2012).