Universidad Zaragoza
1542

Trabajo Fin de Máster

Dynamic and Heterogeneous Wireless Sensor Network for Virtual Instrumentation Services

Autora

Teresa Seco Salvador

Director

Dr. José A. Castellanos

Escuela de Ingeniería y Arquitectura
Departamento de Informática e Ingeniería de Sistemas

2011

Red Inalámbrica de Sensores Heterogéneos para Servicios de Instrumentación Virtual

Resumen

En el presente Trabajo Fin de Master se ha llevado a cabo el desarrollo de un sistema orientado a la adquisición de información sensorial, a través del uso de redes de sensores inalámbricas (WSN, del inglés Wireless Sensor Networks), de un sistema dinámico cuyo comportamiento se desea caracterizar. Para la gestión de la información de los sensores heterogéneos presentes en la red se han aplicado los conceptos de SOA (Service Oriented Architecture) a dicha red inalámbrica, de manera que cada uno de los sensores presentes en la red se trata como un servicio de medida.

La arquitectura propuesta incorpora un mecanismo de "Plug & Play" para la reconfiguración dinámica de la red así como un proceso de composición de servicios que permite la creación de los denominados instrumentos virtuales a través de la asociación de diferentes sensores. Estos instrumentos virtuales agrupan las capacidades de varios sensores heterogeneos de forma que pueden ofrecer al usuario final información de alto nivel complementada con indicios de calidad de dicha información.

Para la obtención de este sistema, las tareas que se han llevado a cabo en este trabajo han sido: se han realizado estudios previos de la utilizacion actual de las redes de sensores inalámbricas y de las arquitecturas SOA aplicadas a WSN. Se ha diseñado la arquitectura de la WSN más adecuada para esta sistema así como el mecanismo "Plug & Play" necesario para el descubrimiento de dispositivos y servicios. Se han estudiado y evaluado los criterios más adecuados para la agrupación de sensores para formar el instrumento virtual de forma automática y transparente. Por último, se ha evaluado la validez de la arquitectura propuesta por medio de su aplicación en un caso concreto en el campo de la logística, en particular, en la supervisión de artículos perecederos. Para ello, ha sido necesario diseñar y definir previamente los módulos de sofware necesarios para la implementación del sistema.

Dynamic and Heterogeneous Wireless Sensor Network for Virtual Instrumentation Services

Sumary

In the present Master Thesis the development of a system oriented towards the sensorial information acquisition through the use of Wireless Sensor Network (WSN) has been carried out in order to characterize the behaviour of a dynamic system. For the management of the information of the heterogeneous sensors present in the network, the SOA (Service Oriented Architecture) concepts have been applied to this wireless network so that each of the sensors can be considered as a measurement service.

The proposed architecture includes a plug & play mechanism for the dynamic reconfiguration of the network and a service composition process which allows for the creation of the so-called virtual instruments through the association of different sensors. These virtual instruments group the capabilities of several heterogeneous sensors in a way that a high-level information with evidence of quality can be provided to the final user.

To obtain this system, the carried out tasks are: previous studies about the current applications of the WSN and about the SOA architectures applied to the WSN have been done. The most adequate WSN architecture for this system as well as the plug & play mechanism necessary for the device and service discovery have been designed. Moreover, the most convenient criteria for the grouping of the sensors to form the virtual instrument in an automatic and transparent way have been studied and evaluated. Finally, the feasibility of the proposed architecture has been evaluated by means of its application in a particular case to the logistic field for the surveillance of perishable goods. For this purpose, the software modules necessary for the implementation of the system have been previously designed and defined.

Contents

1 Introduction **1**

1.1 Objectives . . . 1

1.2 Scope . . . 1

2 State of the Art **3**

2.1 Wireless Sensor Networks (WSN) . . . 3

2.1.1 WSN description . . . 3

2.1.2 WSN applications and drawbacks . . . 4

2.2 Service Oriented Architecture (SOA) . . . 5

2.2.1 SOA Description . . . 5

2.2.2 SOA entities . . . 5

2.2.3 Application to Web Services . . . 6

2.3 SOA model for wireless sensor networks . . . 7

2.3.1 Related approaches . . . 7

2.3.2 WSN constraints for SOA application . . . 8

3 Arquitecture Description **10**

3.1 WSN Elements . . . 10

3.2 Service composition actors . . . 11

3.3 Networking . . . 12

4 SOA for Virtual Instrumentation Services **13**

4.1 Plug & Play Architecture . . . 13

4.2 Composition of Services . . . 14

4.2.1 Configuration Parameters for services composition . . . 16

4.2.2 Services composition process 21

5 Software Architecture 25

5.1 Development Tools 25

5.2 Software modules 26

5.2.1 Acquisition Module 26

5.2.2 Services Module 27

5.2.3 Communication Module 29

5.3 User Interface 29

6 Application to Perishable Goods Surveillance 31

6.1 Hardware description 32

6.2 Services composition examples 33

6.2.1 Example 1: "Temperature alarm when de load temperature exceeds 5 degress Celsius" 33

6.2.2 Example 2: "Conditions of the load during the transport: temperature, vibrations, luminosity and humidity" 34

7 Conclusions and Future Work 37

CHAPTER 1

Introduction

1.1 Objectives

The main objective of this Master Thesis is the design and development of an architecture for the sensorial information acquisition of a dynamic system through a wireless sensor network (WSN). The goal of this acquisition will be the characterization of the system's behavior.

This architecture will enable configuring the WSN in a dynamic and transparent way for the final user, with the incorporation or elimination of sensor nodes to the network by using a plug & play mechanism. These sensors will experience clustering phenomena that will create the so-called virtual instruments.

Each of the heterogeneous sensors present in the network will be treated as a measurement service by means of the application of a Service Oriented Architecture approach (SOA) to the WSN. The aforementioned virtual instruments will emerge from the composition of these services in a transparent manner to the user. As a final result it is intended to get a virtual instrumentation service that it will be accompanied by a quality indicator (Quality of Service index).

Finally, the validation of this method will be tested with a particular implementation of the proposed architecture in a particular case in the logistic fields, specifically in the surveillance of perishable goods.

1.2 Scope

This work has been carried out within the framework of the project Sistema Inteligente de Transporte (SIT) and it has been supported by the Grupo de Investigación Aplicada (GIA-MDPI) from Instituto Tecnológico de Aragón.

In order to reach the proposed objectives the following tasks have been carried out:

- Previous studies of the current use of the WSN and the applicability of the SOA model to the WSN. These studies are described in chapter 2.
- Designing of the WSN architecture which allows creating the proper network to this work. This architecture is described in the chapter 3 of this document.
- Designing of a simple plug & play mechanism and of the services composition process for obtaining the virtual instrumentation services. Chapter 4 describes both the processes and the parameters involved in them in detail.
- Defining the software modules needed for implementing these concepts. The software architecture is described in chapter 5.
- Demonstrating the proposed architecture by means of its application to the perishable goods surveillance. Chapter 6.

The conclusions and the future work are described in chapter 7.

I have been responsible for the development of the main tasks: studying the technologies involved in this work, designing the WSN architecture -including the components selection of the system for the real application-, designing the plug & play mechanism and the service composition process using simulations and defining and implementing the software modules which will be necessary for applying this architecture to the particular case proposed.

As a preliminary result of this work, a paper was accepted and presented in the 8th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2011) that was held in Valencia, in October 2011. This paper is included at the end of the document.

CHAPTER 2

State of the Art

2.1 Wireless Sensor Networks (WSN)

2.1.1 WSN description

Wireless Sensor Networks are computer networks consisting of spatially distributed self-configurable and autonomous sensors. The sensors provide the ability to monitor physical or environmental conditions, such us temperature, humidity, vibration, pressure, etc, with very low energy consumption.

The sensors also have the ability to transmit and forward sensing data to the base station. Most modern WSNs are bi-directional, enabling two-way communication, which could collect sensing data from sensors to the base station as well as disseminate commands from base station to end sensors. The development of WSNs was motivated by military applications such as battlefield surveillance.

A typical Wireless Sensor Network (WSN) is built of several hundreds or even thousands of "sensor nodes". The topology of WSNs can vary among star network, tree network, and mesh network. Each node has the ability to communicate with every other node wirelessly, thus a typical sensor node has several components: a radio transceiver with an antenna which has the ability to send or receive packets, a microcontroller which could process the data and schedule relative tasks, several kinds of sensors sensing the environment data, and batteries providing energy supply.

The main features of WSN are low cost and low energy consumption and fast deployment. Other features are flexibility, scalability, and the ability to provide distributed intelligence.

To reduce cost, each sensor board has very limited onboard resources, such as computing speed, storage and energy source. To achieve long lifetime with limited power supply, usually batteries, onboard components are designed to consume energy as little as possible.

The radio standard most often associated with WSNs is IEEE802.15.4, also known as the Wireless Personal Area Network standard (WPAN). Typical implementations offer a range of up to 300m in open environments, and rates of up to 250 kbits per second. Other radio standards are also competing in this arena. The first is Bluetooth, which is producing a standard called Bluetooth Low Power7 (inheriting from Wibree and Bluetooth Ultra Low Power work). The standard is still under development, but the expected performance is 10m range for a 1 megabit per second rate. The second is low power WiFi implementations (IEEE802.11), which provide a similar range to WPAN but with a significantly higher bandwidth.

Given the ubiquity of the Internet Protocol (IP) in modern communications, there is a growing interest in using it in sensor networks. As the expected bandwidth in these networks cannot support the overhead of IP, stateless compression techniques have been devised in the Internet Engineering Task Force (IETF), with IPv6 over Low power WPAN (6LoWPAN). The resulting overhead is appropriate for WPAN technology, but requires a node to compress / decompress, as well as fragment, due to packet size restrictions.

The standards described so far provide generic networking capability, limited to the lower layers of the networking stack. A number of significant standards have emerged which cover higher layers. ZigBee is built on top of IEEE802.15.4, and adds networking, transport and application profile layers [3].

2.1.2 WSN applications and drawbacks

The increasing and rapid development of WSN has facilitated their application in different fields such as environmental surveillance [1], healthcare monitoring [22], home automation, structural health monitoring [10], logistics [4], etc.

However, sensor network architectures are tailored to specific applications and in most cases are device dependent. For these reasons it is necessary an abstraction layer for supporting the interoperability between different applications. This abstraction layer is obtained through the application of Service Oriented Architecture model to the heterogeneous wireless sensor network.

In the next chapters a description of SOA and its application to WSN are described.

2.2 Service Oriented Architecture (SOA)

2.2.1 SOA Description

Service-Oriented Architecture (SOA) is an architectural concept for designing and implementing distributed systems such that functionality is encapsulated into interoperable services. A service is an implementation of a clearly defined self-contained function that in principle operates independent from the state of any other service. It has a well defined set of platform-independent interfaces and operates through a pre-defined contract with the consumer of the service. Services are loosely coupled and all interaction takes place through the interfaces.

SOA is a logical way of designing software systems to provide services either to end-user application or other services distributed in a network, via published and discoverable interfaces.

2.2.2 SOA entities

The SOA is based upon the interactions between three roles: service provider, service registry and service requestor. The interactions involve the publish, find and bind operations.

In a typical scenario, a service provider hosts a network-accessible software module (an implementation of a service). The service provider defines a service description for the service and publishes it to a service requestor or service registry. The service requestor uses a find operation to retrieve the service description locally or from the service registry and uses the service description to bind with the service provider and invoke or interact with the service implementation. Service provider and service requestor roles are logical constructs and a service can exhibit characteristics of both. Figure 2.1 illustrates these operations, the components providing them and their interactions.

Then, the SOA model consists of the following entities configured together to support the find, bind, and execute paradigm:

- Service Consumer: The service consumer is an application, service, or some other type of software module that requires a service. It is the entity that initiates the

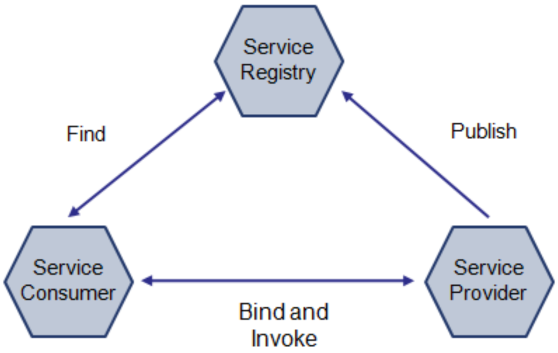

Figure 2.1: SOA entities and theier operations

locating of the service in the registry, binding to the service over transport, and executing the service function

- Service provider: The service provider is the service, the network-addressable entity that accepts and executes requests from consumers. It can be a mainframe system, a component, or some other type of software system that executes the service request. The service provider publishes its contract in the registry for access by service consumer.

- Service Registry: A service registry is a network-based directory that contains available services. It is an entity that accepts and stores contracts from service providers and provides those contracts to interested service consumers.

2.2.3 Application to Web Services

SOA is a concept which is not tied to a particular technology. However, Web Services are currently the preferred framework to deliver interoperable SOA. In Web Services-based SOAs, the contract is defined by a WSDL (Web Services Description Language) document, which stipulates how service consumers can bind to a service producer by exchanging messages using a defined XML (Extensible Markup Language) grammar. The main protocols that Internet services use today is SOAP (Simple Object Access Protocol) and REST (Representation State Transfer) and the most commonly used registry model in web services is based on Universal Description, Discovery and Integration (UDDI) specification. For orchestrating services it is used the Business Process Execution Language for Web Services (BPEL4WS). Figure 2.2 shows the web services architecture.

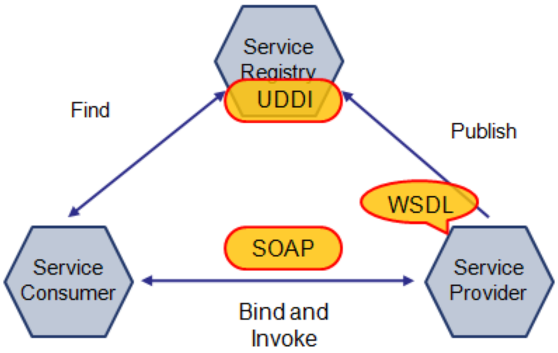

Figure 2.2: Web Services architecture

2.3 SOA model for wireless sensor networks

2.3.1 Related approaches

The Service Oriented Architecture (SOA) has been considered as a good candidate to develop open, efficient, inter-operable, scalable and customizable WSN applications. By wrapping application functionality into a set of modular services, a programmer can then specify execution flow by simply connecting the appropriate services together.

One of the earlier works in taking the service-oriented approach in the design of a middleware system for WSN was presented in [6]. In this work it is introduced a simple service-oriented model in which the responsibility for handling the services requests is assigned to an external entity which acts as a bridge between the requests received from the exterior and the internal network functionality.

In [18] the authors propose a service oriented architecture consisting of service composition layer on top of a basic service oriented architecture called ESOA (Extended Service Oriented Architecture). This service composition layer encompasses the necessary roles and functionality for the consolidation of multiple services into a single composite service. Through this composition layer it is possible to perform functions like the coordination to control the execution of the component service, monitoring to allow subscriptions to events produced by the component services and conformance to ensure the integrity of the composite service. The ESAO are implemented on a LiteOS, a new operating system for wsn.

TinySOA [16] is a prototype SOSANET (Service Oriented Sensor Actuator Network) developed on top of TinyOS. In this system services are lightweight code units deployed directly on top of the operating systems of nodes. Applications invoke services using a service-oriented query model. Queries are submitted to one of the established base station or directly to the individual node.

In [9], it is presented OASiS, an object-centric, ambient-aware, service-oriented sensor network programming framework and middleware. In this approach, a physical phenomenon of interest is represented by a finite state machine (FSM) referred as logical object, which is assigned to a particular node (object node) after a creation protocol. OASiS includes a dynamic service configuration mechanism for the integration of the real world (communication failure, node dropout, etc.). This middleware comprises a set of services which include a node manager, a composer, a service discovery protocol and a object manager.

Atlas [7] is a service-oriented sensor platform with middleware based on OSGi (Open Service Gateway initiative) that enables programmable pervasive spaces. Atlas nodes, modular hardware platforms with sensor/actuator, microprocessor and communication layer, provide only limited processing power and most of the service functionality is executed on a stand-alone server where Atlas nodes register.

TinyWS [14] is a small web service platform that resides on the sensor nodes. It hosts the web services and has a SOAP processing engine. The sensor nodes are service providers, the application devices are service requestors and a distributed UDDI acts as an overlay entity. These platforms permits the direct interaction between applications and individual sensor nodes for sinkless WSN.

In [5] the authors present RASA (Resource Aware Service Architecture). This service architecture uses services which are injected to the network by a node (inquirer) at runtime. A service is forwarded to other nodes and installed as well as executed on all nodes of the network successively. After their installation, services are accessible by other nodes and other services, too. Due to small available data memory in sensor nodes and the relatively high energy costs of data memory compared to flash memory, they propose to store the code and most of data in flash memory.

2.3.2 WSN constraints for SOA application

The Service-Oriented Architecture (SOA) can bring enormous benefits to the WSNs because it can turn these networks into open, ubiquitous, interoperable, and multipurpose infrastructures. The programming task becomes easier because sensor node capabilities are abstracted and defined as services and applications are written based on service requests issued to the network.

The programming task becomes easier because sensor node capabilities are abstracted and defined as services and applications are written based on service requests issued to the network. However, sensor nodes are usually quite restricted in resources (memory size, processing power, battery power). This resource-constrained nature of sensor nodes has negatively influenced the proposed SOA solutions, since the majority of them rely on in-the-middle resource rich devices to make the bridge between the outside world and the WSNs.

Moreover, XML format is extremely verbose and its processing consumes a significant amount of time and memory and the bandwidth usage is relatively high for wireless sensor networks. For these reasons, we propose a simple protocol and software that reproduces the architectural concepts and information exchanges of SOA implementations.

One of the main goals of the proposed work is to make sensor nodes capable of hosting services, announce them in the network and discover other services, so the system should include a service and device discovery protocol. The service composition plays and important role in systems which intend to work with heterogeneous devices formed by cluster of sensors. For this reason the proposed system includes a services composition method for the creation of these clusters that we call virtual insturments.

CHAPTER 3

Arquitecture Description

The main goal of the proposed architecture is to accomplish a universal platform which permits plug & play sensor nodes connection to the network by means of which it is possible to configure different measurements services.

3.1 WSN Elements

This system, based on wireless sensor networks, is comprised of three elements: the sensor nodes, the master node and the control centre.

The sensor node is the network entity in charge of the data acquisition of the sensor or sensors connected to the node. It includes communication capabilities used to send the information both to other sensor nodes and to the master node. The sensors attached to the node measure heterogeneous magnitudes not only physical (temperature, humidity) but also logical (RFID for identification purposes). There are different types of sensor nodes depending on its processor capabilites. This node acts as service provider or service consumer depending on the task.

The master node is the main network node and its main functionalities are forming and maintaining the network and recognizing the sensor nodes present in the network and the services provided from them through a plug&play mechanism (device and service discovery). Moreover, it is responsible for the service composition process through the selection of a sub-master. The master node will act as service consumer and as service registry. This node communicates with the sensor nodes, with other distributed master nodes and with the control centre.

The sub-master is a singular node for every requested service, either for proximity to the master node or because it is a special type of sensor needed for that service. This special node, named sub-master, is the main node of the virtual instrument. The main task of the sub-master is to coordinate the rest of the sensors that form the virtual instrument. This sensor node plays the role of service master for the required service.

Finally, the control centre is in charge of gathering all the network information and provides a user interface for the high level service request, so it acts as a service registry and as a consumer of high-level services.This information can be accessed by other high level networks (IoT).

With all the information about the wireless sensor network, the user can ask for a high-level service that could involve heterogeneous devices present in the network. In order to respond to this query it is necessary the existence of a services composition method.

Figure 3.1 shows all the elements of the wireless sensor network.

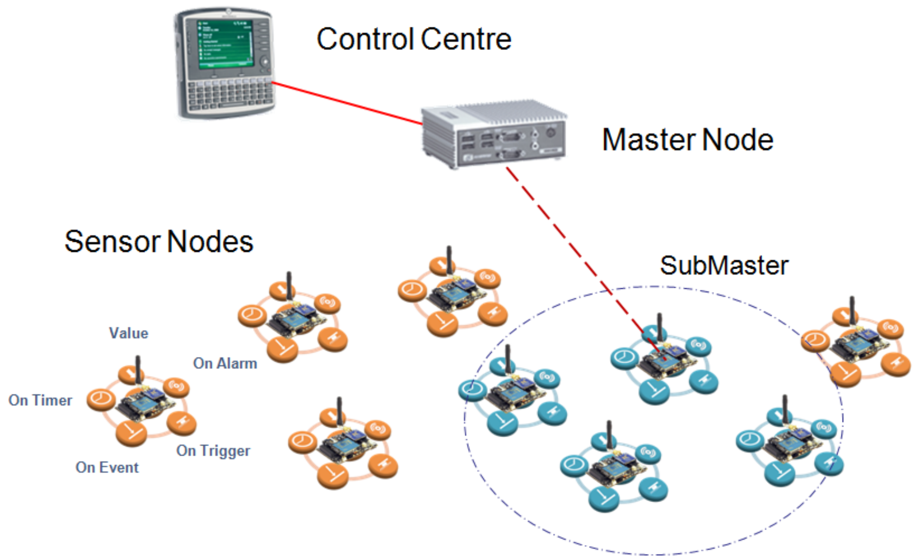

Figure 3.1: WSN elements

3.2 Service composition actors

To understand the proposed composition of services we need to identify some concepts: Homogeneous sensor Array (HoS) is defined as the entity which provides a homogeneous measure. This measure value is obtained from a set of sensors which return the same physical (temperature, humidity, light sensor) or logical (RFID, presence) magnitude. The association of different homogeneous sensors is what we call heterogeneous sensor array (HeS) or virtual instrument (VI) (Figure 3.2).

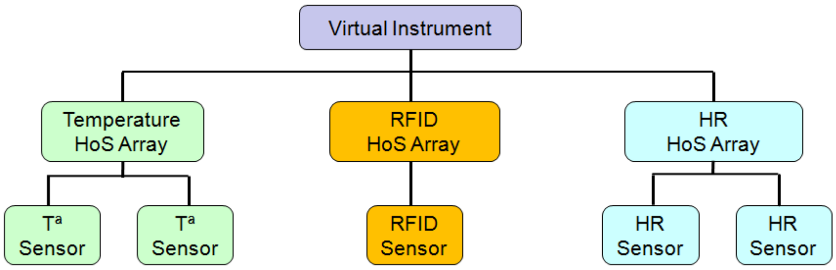

Figure 3.2: Actors in services composition process

3.3 Networking

Considering the features and requirements of the system, wireless sensor networks (WSNs) have been selected as the best option to integrate all components because of their easiness of use, low-cost and low power consumption. In the proposed architecture it is necessary that the sensor nodes are able to directly communicate among them, in order to create the necessary associations that make possible to form the virtual instruments. For this reason the standard chosen is the IEEE 802.15.4 (level 2 in the OSI model), which enables point-to-point communication and does not imply the existence of a network coordinator, as happened with standard Zigbee (level 3 and uppers in the OSI model). Consequently, a mesh topology has been implemented for the WSN. The band frequency used is the 2.40-2.48GHz band (16 channels and 250 Kb/s bit rate).

It is possible to use different type networks like GPRS, UMTS, WLAN, etc. for communication between the master node and the control centre.

CHAPTER 4

SOA for Virtual Instrumentation Services

The SOA-based architecture proposed involves two main stages: services and device discovery through a plug & play mechanism and the composition fo services to form virtual instrument services.

The initial scenario consists of an array of sensor nodes with different environmental and logical sensors attached and a master node. In a first step, the sensor nodes and their services are discovered through the implemented plug & play protocol, and all this information is gathered and available in the master node and in the control centre. At a second stage, the user defines the requiered high level service through the control centre and the master node makes a query to the network for this service; at this moment the proposed services composition comes into play with the selection of the sensors which are going to form the virtual instrument.

In the next subchapters both the plug & play architecture and the services composition method are described.

4.1 Plug & Play Architecture

This feature allows users to know the capabilities of the nodes in the network. For this purpose, each sensor node contains information about its main features. Part of this information correponds to factory settings and the other part is assigned to each sensor node in an initial configuration process. The node information is organized in different data structured levels:

- Data Level 0: sensor intrinsic parameters like type of measurement, sensing unit, sensing range, sensitivity, offset, calibration information, etc.

- Data level 1: unique node identification (nID), list of sensors attached to the node, topological location of the node (locID), etc
- Data level 2: description of available services, events information (event list, type of event, thresholds, etc.) and communication message formats for interfaces.

These data structures provide the nodes with the ability to self-describe their features and available services so that the users have the knowledge required to interact with them.

A services and device discovery protocol has been implemented based on message passing. This protocol allows available sensor nodes to be registered to the master node. The master node starts the process by sending a broadcast message -which is called "whoIsAlive"-, to the network. Then, the sensor nodes present in the network respond to the master with an "alive" message. This message includes the node identification and the network address. Using this information, the master node retrieves the configuration data from the discovered sensor node through a query message "getSensorConfig". When a sensor node receives this query, the response is built based on the information contained in the data structure and returned to the master ("sensorConfig" message). These data include the number of sensors attached, the meassure type, the topological location, the available services, etc. In a similar way, when a a sensor node joins the network sends an "alive" message and with the information contained in this message the master is able to obtain the internal data of this new node through the "getSensorConfig" message (Figure 4.1).

The master node mantains a list of the sensor nodes and their main features which are present in the network. This dynamic table is continously updated by means of "whoIsAlive" messages that the master node periodically sends to the network.

At this point, the necessary information for creating virtual instruments is available in the network and it is possible to afford the stage of services composition.

4.2 Composition of Services

The goal of services composition process is to provide the high-level services requested by the user in a transparent manner. There is a list of available heterogeneous sensors that have been discovered through the plug & play protocol decribed before. All this

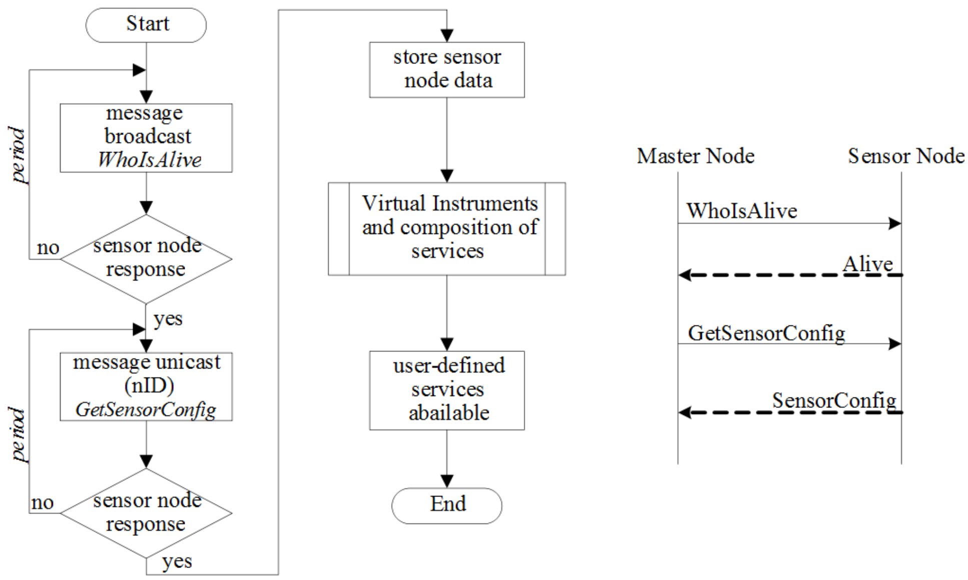

Figure 4.1: Plug & Play mechanism

information about sensors are shown to the user in form of templates, so the user is able to select a high-level service. At this moment, the master starts the service composition process. When this process ends, the master and the control centre have the information about the virtual instrument formed and then they can run the service.

Given a collection of heterogeneous sensors we can define a rearrangement of the items according to the user preferences and the available resources. Each user-defined service is formed by one or more groups of homogeneous measures (HoS). The service resulting from the association among homogeneous sensor (HoS) is the measurement service (MS). The association of different homogeneous measures forms the virtual instrument.

The associations both between homogeneous sensors and heterogeneous sensors are carried out in base on a set of configuration parameters that the user selects when asking for the service.

4.2.1 Configuration Parameters for services composition

4.2.1.1 Configuration parameters for homogeneous sensor array

The result of the associations between homogeneous sensors of the same family is presented as the measurement service. The configuration parameters that are involved in the selection of the sensor nodes of the homogeneous measure are:

- Family Sensor (*FS*): Variable that represents the class of the magnitude measure by the sensor, for example: temperature, humidity, luminosity, etc.
- Number of Measures (*N*): Number of sensors required of the family
- Statistical Treatment (*ST*): Statistical (or sensorial fusion) function to deal with the homogeneous array of measures. The goal of this parameter is to obtain a single measurement from the set of sensors. The selected function for this purpose is described in subchapter 4.2.1.2
- Topological Location Identification (*locID*): An identification which links the sensor with a topological location. This parameter is associated to the sensor node in a initial stage of configuration. In a topological representation, unlike a metrics representation where the position refers to one of the coordinate axes, the environment is divided into areas with similar features. In this case, the topological division is based on the features of the sensor nodes present in the network. Moreover, in the proposed services composition process, it is not necessary a high precision localization and therefore it is enough with a topological location parameter.
- Behaviour (*BH*): It describes the situation or event that generates a response from the set of the homogeneous sensors. This parameter describes the available simple service attached to the each sensor. Table 4.1 shows some examples for this parameter.
- Spatial dispersion (σ_S): Integer and dimensionless dispersion unity of homogeneous sensors with reference to the sub-master of the service. This parameter is described in 4.2.1.3 subchapter in more detail.

BH	Description
OnAlarm	Return the value when alarm appears
OnPeriod	Return the value periodically
OnRequested	Return the value when it is available
OnEvent	Return the value when a sensor event occurs
OnWindow	Return the value when it is within a range window
OnTrigger	Return the value when a trigger signal appears

Table 4.1: Behaviour parameter

- Trigger (*Trigg*): Origin of a trigger signal (if triggered). It is strongly linked with the behaviour parameter. It is a type of special behaviour consisting of a direct message from the trigger node to another node when a condition appears.

4.2.1.2 Statistical treatment for homogeneous measures

As defined above, the statistical treatment parameter refers to a statistical or fusion function in order to deal with the homogeneous arrays of measures. The goal of this function is to obtain a single value from the set of observations (measures) of the same family discarding the outliers. Due to the nature of the observations, it has been selected a median filter as the fusion function.

The standard median filter is a simple nonlinear smoother very robust that can suppress noise while retaining sharp sustained changes in signal values. It is particularly effective in reducing impulsive-type noise.

The output of median filter at a point is the median value of the input data inside the window centered at the point. If is $\{x(k)|\, 1 \leq k \geq L\}$ and $\{y(k)|\, 1 \leq k \geq L\}$ are, respectively, the input and output of the one-dimensional (1-D) median filter of window size 2N + 1, then:

$$y(k) = median\,\{x(k-N), ...x(k-1), x(k), x(k+1), ...x(k+N)\} \tag{4.1}$$

Median filter is a nonlinear signal processing tool which offers advantages in applications in which the underlying random processes are nonGaussian. Practice has shown that nonGaussian processes do emerge in a broad array of applications, including wireless communications, teletraffic, hydrology, geology, economics, and imaging.

The common element in these applications, and many others, is that the underlying processes of interest tend to produce more large-magnitude (outlier or impulsive) observations than those that would be predicted by a Gaussian model. That is, the underlying signal density functions have tails that decay at rates lower than the tails of a Gaussian distribution. As a result, linear methods which obey the superposition principle suffer from serious degradation upon the arrival of samples corrupted with high-amplitude noise. Nonlinear methods exploit the statistical characteristics of the noise to overcome many of the limitations of the traditional practices in signal processing [2].

In this case, the input data are the discrete measures associated with spatial points which correspond to the sensor selected for the homogeneous measurement. Accordingly, the representative value of this homogeneous measurement is obtained through the simple application of the statistical median to the discrete measures.

4.2.1.3 Spatial Dispersion

This parameter describes the dispersion desired for the homogeneous sensors with reference to the sub-master of the service. It is used to select the nodes which are going to be involved in a particular service requested by the user, assuming a homogeneous distribution of the sensors.

It is based on the Received Signal Strenght Indicator (RSSI) of the last received RF data packet in dBm. This indicator is used as a distance estimator because it has some advantages in contrast to other distance estimator techniques like ToA/TDoAe (Time of Arrival/Time Different of Arrival) or AoA (Angle of Arrival) [11]. This technique uses the attenuation properties of the radio signal for modelling the distance between two nodes. The values for RSSI ranges from 0 to 255 where zero value corresponds to the maximum coverage of the radio module used, for example, the range of XBee 802.15.4 RF Module is up to 30 meters so zero value corresponds to this value.

Although the RSSI is sensitive to noise and has some inherent problems in the waves propagation like multipath fading, shadowing or path loss, it provides good performace for short distances with an adecuate data postprocessing. The main advantage of the use of this distance estimator is that additional hardware is not necessary because the value of RSSI is included in the received data frame. Moreover it has no impact in power comsumption, sensor size and thus cost.

The spatial dispersion value for sensor node i is estimated by equation 4.2.

$$\sigma_{Si} = floor\left(\frac{255 - RSSI_{sm,i}}{25.5} + (N_h - 1)\frac{1}{n}\sum_{j}^{n}\frac{255 - RSSI_{sm,j}}{25.5} + 0.5\right) \quad (4.2)$$

where $RSSI_{sm,i}$ is the RSSI of the last packet of sensor i received by the sub-master, n is the number of nodes with direct connection with the sub-master (without multi-hops) and N_h is the number of hops that a particular node needs to communicate with the submaster when there is no direct connection between them.

The figure 4.2 ilustrates this expression through an example:

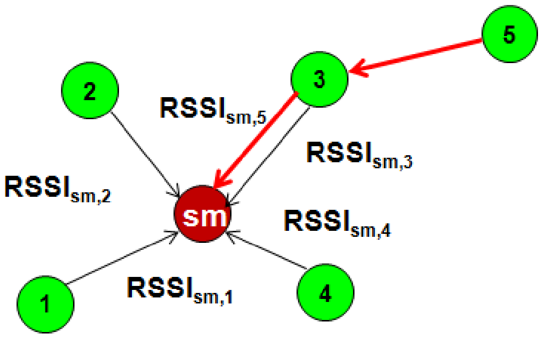

Figure 4.2: RSSI and hops

In this example, the sensor nodes 1, 2, 3 and 4 have direct connection with the sub-master so the number of hops for all of these nodes will be one. Therefore, only the first term of the equation is involved in the spatial dispersion calculation. The sensor node 5 needs two hops to communicate with the submaster. In this case, the two terms of the equation are involved in the spatial dispersion calculation for this node.

The maximum value of this parameter depends on the distribution of nodes over the considered area. A result between 0 and 10 indicates that most of the sensors are in direct communication with the sub-master; if the result is 10 to 20, the interested area also contains nodes with one hop, and successively.

When a user asks for a high-level service, he selects an spatial dispersion range depending on the distribution of the sensors requiered. The graphic represented in figure 4.3 shows this situation.

4.2.1.4 Measurement Service and Quality of Service Index

The measurement service is obtained through the expression:

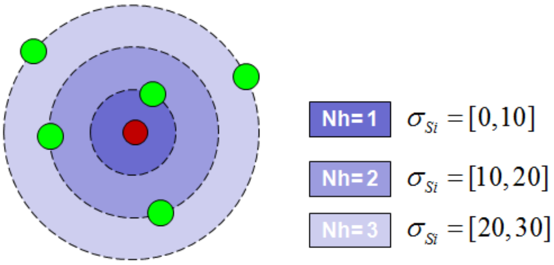

Figure 4.3: Spatial Dispersion Ranges

$$MS = \begin{pmatrix} V \\ t \\ QoS_{MS} \end{pmatrix} = f\left(FS, N, ST, locID, BH, \sigma_S, Trig\right) \quad (4.3)$$

When the measurement service is formed in base on the configuration parameters, it is possible not to fullfill all the user requierements, but the service can likewise be given. To take into account this situation, a quality of service index has been added . This index is an indicator of the quality intended to be a measure of the distance from the requested service to its actual deployment, in other words, the matching between the user requirements and available resources is measured by this QoS index.

Therefore, the result of the measurement service is a value of the measure in a timestamp and a QoS index. The calculation of this index is done with the equation:

$$QoS_{MS} = 1 - \frac{N - m}{N} \quad (4.4)$$

where N is the number of required sensors and m is the number of selected nodes that meet the condition specified by the spatial dispersion parameter σ_S. The QoS index value ranges from 0, where there is no sensor which satisfies the specificacion, to 1 where there are enough sensors which satisfy the user requeriments.

4.2.1.5 Virtual Instrument Service

As has been mentioned before, the association of different homogeneous services forms the virtual instrument. Therefore, the virtual instrument service depends on the collection of measurement services that are involved in the virtual instrument (MS_i). The expression for obtaining the virtual instrument service VS is represented by the equation 4.5:

$$VS = \begin{pmatrix} LV \\ t \\ LQoS \end{pmatrix} = f\left(MS_1 : MS_2 \cdots MS_n, f_{fusion}, BH_{VI}\right) \quad (4.5)$$

The result of the virtual instrument service could be a list of values corresponding to each value of measurement service or a single value estimated by using a fusion algorithm represented by the f_{fusion}. Finally, there is another parameter, BH_{VI}, which describes the event that generates a response from the VI. This parameter is similar to the behaviour parameter described in the measurement service case. Values for this parameter could be those shown in the table 4.1.

The resultant list of values are accompained by a list of Quality of Service indexes of the individual measurement services. This information is useful for identifying the measurement service responsible for not fulfilling the user requirements. Additionally, a global index is provided to the user represented with this equation:

$$QoS_{VS} = \frac{\sum_{1}^{n} QoS_{MS_i}}{n} \quad (4.6)$$

where n is the number of measurement services which forms the virtual instrument.

4.2.2 Services composition process

The initial situation for this process is a list of available heterogeneous sensors with their main features that have been discovered through the plug & play protocol. This list is present both in the control centre and in the master node. The user makes a high level service query and then the master node starts the service composition process to obtain the required service, that is to say, the virtual instrument service.

This process involves two main sub-processes, the sub-master selection which is executed by the master node, and the selection of the sensors that are going to form the virtual instrument. This last sub-process is executed by the sub-master node.

The configuration parameters described before are involved in both sub-processes. The next points explain the sub-master selection and the sensors selection in detail.

4.2.2.1 Sub-master selection process

In order to select the appropriate criteria for the sub-master selection it has been developed a simulation program in Matlab. With the results of the simulations an algorithm has been design.

The input parameters for the sub-master selection algorithm are the configuration parameters selected by the user when he asks for the service. These parameters are:

- Number of different magnitudes involved in the service
- For each magnitude: type of the magnitude *FS*, number of sensors requested *N*, *locID*, spatial dispersion range $[\sigma_{\min}, \sigma_{\max}]$ and behaviour desired *BH*. These parameters correspond to the parameters involved in each measuremente service.

The number of messages requiered for each process has been taken into account in these simulations in order to select the criteria which adds the lower overhead to the communication.

The criteria followed to select the sub-master are sumarized as follows: if there is an only sensor for a particular family of sensors then it is selected as the sub-master because this node is essential for the service. If this condition is not satisfied, then the centroide sensor node of the candidate sensors is selected as the sub-master. This criterion is selected because the centroide sensor is the sensor with best quality of communication with all the candidate sensors. To calculate the centroide it is necessary to compute, for each node, its distance to the rest of nodes. For this purpose the RSSI value is used as distance estimator. It is possible that more than one node has similar distances. In this particular case, the candidate sensor near the master node is selected as the sub-master.

Therefore, the implemented algorithm includes the following steps:

1. There is a list of requested magnitudes of different families *FS*
2. For each *FS*, a list of sensors with the same *locID* as requested is selected -> list of candidates
3. The master node computes and stores the number of candidates for each *FS*: nc_i
4. $nc_i = 1$?

(a) Yes -> sensor i is selected as sub-master: s_i

(b) No -> Another criterion is needed

- Computing the distances among candidates -> d_{ij}
- The sensor which is nearest the other candidate sensors is selected as sub-master:
  $s_i \rightarrow \min\left(\sum_{\substack{j=0\\j\neq i}}^{N} d_{1j}, .., \sum_{\substack{j=0\\j\neq N}}^{N} d_{Nj}\right)$
- If there are some sensors with similar distances the candidate sensor nearest the master is selected:
  $s_i \rightarrow \min\left(d_{mi}\right)$

4.2.2.2 Sensors selection process

As a result of the previous process, there is a sub-master node responsible for the service and a list of candidate sensors for each family sensor, in other words, for each measurement service. This information is available both in the master and in the sub-master node which is in charge of the selection of the sensors among the candidate ones that are going to form the virtual instrument.

As in the previous case, a simulation program has been developed to obtain the criteria used for the selection of the sensors.

In this second process some of the configuration parameters selected by the user are involved: the number of sensors required for each FS (*N*) and the spatial dispersion range $[\sigma_{\min}, \sigma_{\max}]$. With these parameters and the list of candidate sensors the submaster is able to starts this process. In a first step, the sub-master calculates the spatial dispersion of each candidate sensor relative to the sub-master. Only the candidate sensors with an spatial dispersion value within the user selected range are added to the list of selected sensors for each magnitude. In a second step, the QoS index is calculated using the number of selected sensors and the number of required sensors. If this value is equal to one, it implies that the sensor nodes available in the wsn are enough for satisfying the service requested by the user. If this value is lower than one it implies that the service can be executed but not with all the resolution desired. There is a third case where the QoS index is higher than one. The meaning of this situation is that there are more sensors than the requested ones which satisfy the user requirements. Accordlingy it is necessary to select another criterion like to make a random selection between the selected sensors. This criterion has been chosen in order to avoid spending time and memory in other types of calculations.

Therefore, the implemented algorithm for sensors selection includes the following steps:

1. Initial situation: sub-master sm, candidate sensors cs_i, number of required sensors nr_{MSj} and spatial dispersion range $[\sigma_{\min}, \sigma_{\max}]$.
2. The sub-master computes $\sigma_{sm,i}$
3. $\sigma_{\min} \leq \sigma_{sm,i} \leq \sigma_{\max}$?
   (a) Yes -> add s_i to the list of selected sensors
4. The sub-master stores the number of selected sensors for each MS: ns_{MSj}
5. QoS calculation:
   $QoS_{MSj} = 1 - \frac{nr_{MSj} - ns_{MSj}}{nr_{MSj}}$
   - if $QoS_{MSj} = 1$ then process ends: ok
   - if $QoS_{MSj} < 1$ then process ends -> quality of service depends on value
   - if $QoS_{MSj} > 1$ then random selection -> process ends -> ok

At the end of both processes, sub-master and sensors selection, the virtual instrument is built and then it is possible to configure the virtual instrumentation service with the desired behaviour (*BH*) and to make the bindings between the selected sensors and the submaster. Finally, the service can runs.

CHAPTER 5

Software Architecture

5.1 Development Tools

As has been mention before, there are different hardware platforms for the sensor nodes and for the master node. For environmental sensors which do not need high processor capabilities, the Waspmote platform from Libelium has been selected. For other types of sensors which need more processor and memory capabilities an EBOX platform with an Intel Atom processor has been selected.

The selection of the software development tools and the programming language has been conditioned by the hardware platform used and by the role of each device (master, sensor or control centre).

The software development tools used are:

- Waspmote IDE V0.1 and C language for sensor nodes using the Waspmote platform. Waspmote API libraries are provided with this IDE.
- Visual Stuido 2005 and C++ for nodes (both sensors or master) using the EBOX platform and for the user interface application.
- Visual Studio 2005 and C# for control centre.

Several applications have been developed for the implementation of the functionalities of the nodes which form he system:

- Master node application: This application is responsible for the following tasks:
  - Sending and receiving the RF data packets through XBee Radio module using the serial port
  - Decoding of the messages

  - Maintaining the list of the active nodes and their features.
  - Executing the process for sub-master selection
  - Communicating with control center through ethernet

- Sensor node application: This application is responsible for the data acquisition of the sensor, for the pre-processing of these data and for the messages creation to respond to a service query.

- Control Centre application: This application is responsible for providing a user interface for the high-level service queries.

5.2 Software modules

The software necessary for implementing the functionalities of the nodes of the network consists of different modules: acquisition module, communication module and service module.

The synchronization between modules and the scheduling of the rest of tasks is carried out by a main process executed by the microprocessor.

5.2.1 Acquisition Module

This software module is in charge of the signal acquisition of the sensors and it is responsible for the initial processing of the acquired data.

The environmental sensors (temperature, luminosity, HR and vibrations) are attached to a Waspmote platform. The values of these sensors are obtained by the analog inputs readings by means of a function "analogRead()" provided by the WaspmoteAPI.

For the acquisition of the data of other types of sensors, like 3D volume sensor or RFID reader, the development of middleware applications is necessary. These applications are in charge of the following tasks:

- Data acquisition using the adequate drivers for handling the communication with each sensor (USB, Ethercat, etc.)

- Data processing in order to obtain a singular value which represents the magnitude measured by the sensor; for example, it is necessary to process the image data returned by the 3D sensor to obtain a volume value.

5.2.2 Services Module

This module is responsible for building the user frames which include the necessary information for handling the services. For this purpose, a user frame has been designed to include all the needed information for the plug&play protocol and for the service composition process. The next table shows the implemented frame format:

2b	**1b**	**1b**	**1b**	**1b**	**1b**	**MAX_DATA-8**	**1b**
idCod	Fr ID	H	M	S	len	data	cks

Table 5.1: User frame format

where:

- idCod: Identification code of the function
- Fr ID: Frame Identification for synchronism purpose
- H,M,S: Hour, minute and second
- len: Data length without cks
- data: Data to send or receive. MAX_DATA is the maximum size of the data frame. This value will depend on the maximum payload allowed for the XBee frame.
- cks: Checksum

Tables 5.2 and 5.3 show some examples of functions used in the communication between master and sensor node.

Different identifications for measure and service type have been defined. Table 5.4 and 5.5.

idCod	FUNCTION	SOURCE	DESTINATION	PARAMETERS	SIZE PARAM	SENSOR RESP
0x04	WhoIsAlive	master	broadcast			Alive
0x13	GetSensorConfig	master	sensor node			SensorConfig
0x31	ValueNow	master	sensor node	idMeasureType		DataValue

Table 5.2: Master to sensor node

idCod	FUNCTION	SOURCE	DESTINATION	PARAMETERS	SIZE PARAM
0x06	Alive	sensor node	master		
0x09	SensorConfig	sensor node	master	numMeas idMeasureType_0 valueSize numServices_0 idServiceType_00 idServiceType_n0 idMeasureType_n valueSize numServices_n idServiceType_n0 idServiceType_nn	1 1 1 1 1 1 1 1 1 1 1

Table 5.3: Sensor to master node

idMeasureType	Magnitude	Value Size	Type
0x01	Temperature	2	Analog
0x02	HR	2	Analog
0x06	Volume	2	Eth driver
0x50	Presence	1	Digital
0xA0	CDB	variable	usb driver
0xA1	RFID	variable	usb driver

Table 5.4: Measure types

idServiceType	Service	Parameters	Param Size	Description
0x01	ValNow			Current value
0x02	ValRequest			Value when available
0x03	ValOnPeriod	Period	2	Value periodically
0x05	ValOnTrigg			Value when trigger input
0x08	ValOnAlarm	limInf limSup	2 2	Value when alarm

Table 5.5: Service Types

5.2.3 Communication Module

This module is in charge of data communication through the XBee RF module. It builds the 802.15.4 XBee frames ([19],[21]), adding the user frame constructed by the service module to it. Communication with the XBee module is carried out through the UART, in the Waspmote case, and through the serial port in the case of the EBOX, using an XBee adapter.

The XBee frames are shown in the following tables:

1b	2b		1b
SD	LEN	FRAME DATA	CKS

Table 5.6: 802.15.4 Frame format

1b	1b	64 bit MAC Destination Addr	1b	
API ID	FR ID	add0 .. add7	OPT	RF DATA

Table 5.7: Frame Data: 64 bits MAC Destination Address

1b	1b	1b	1b	64 bit MAC Source Id Origin node	
APP ID	Frag ID	#	ST ID	add0 .. add7	USER DATA

Table 5.8: RF Data: 64 bits MAC Source Address

5.3 User Interface

A basic graphical user interface has been developed for obtaining the information about the nodes available in the network and their configuration as well as for requesting basic services.

This interface has been implemented inside the master node application and it is oriented towards the validation of the functionalities of the system rather than a user interface for the final user. Moreover, another application has been developed in the control centre which gains access to the information provided by the master. This last application is not part of the frame of this work.

Figure 5.1 shows the graphic user interface.

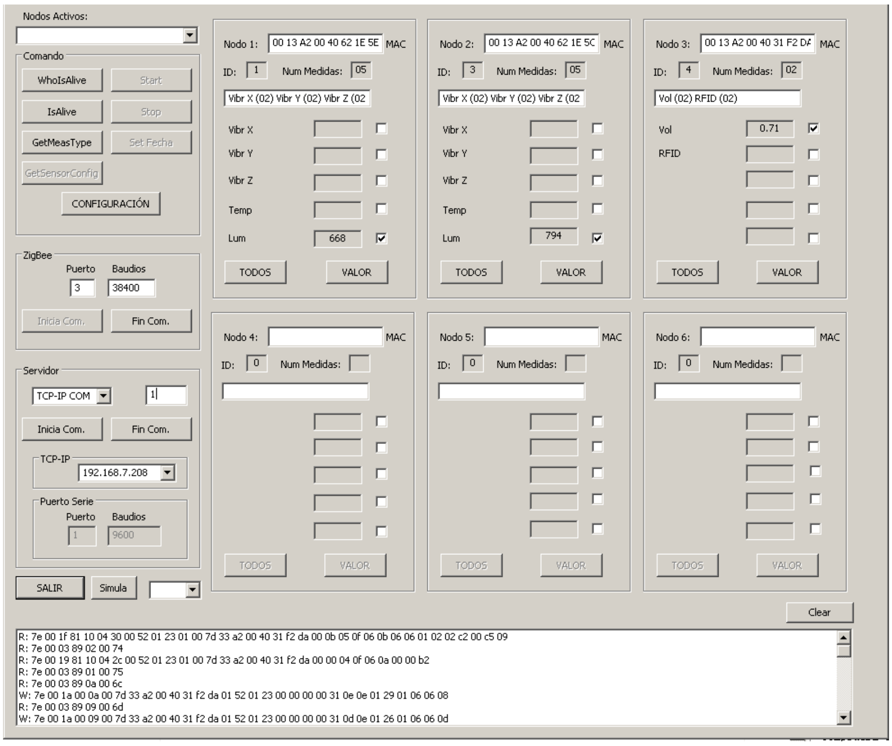

Figure 5.1: Graphic User Interface

CHAPTER 6

Application to Perishable Goods Surveillance

In order to show the usability of our method, the proposed architecture has been implemented in an intelligent transportation system setting where both surveillance and control tasks of perishable goods are required.

The suitability of WSN for food transport monitoring has been demonstrated in other works: In [4] it has been deployed mobile WSN in a cargo container on a trans-Atlantic cargo vessel as well as on a lorry to monitor the transport conditions inside the container. In [17] the use of two types of wireless nodes for monitoring storage and transport was experimentally assessed.

Perishable goods decay rapidly and must be therefore transported by highly efficient distribution channels that can retain the integrity of the product. The objective of the experimental application of this architecture is to control the biological degradation process of perishable foods by means of the measurement of some critical parameters during transportation:

- Temperature: brief interruptions in the control of the cold chain can result in an immediate deterioration of product quality.
- Light intensity: this value is checked with the aperture of the door, and therefore with exposure to external agent
- Humidity: to detect leaks of water in the product.
- Vibrations: this parameter affects the quality of some vegetables.

In order to measure these magnitudes, the sensor nodes have been distributed inside the truck container. To demonstrate the use of this architecture with other type of sensor nodes with different computing capabilities, two more sensors have been added to the system: a RFID reader sensor for identification of the packages and a 3D volume sensor for the measurement of the occupancy level of the load (Figure 6.1).

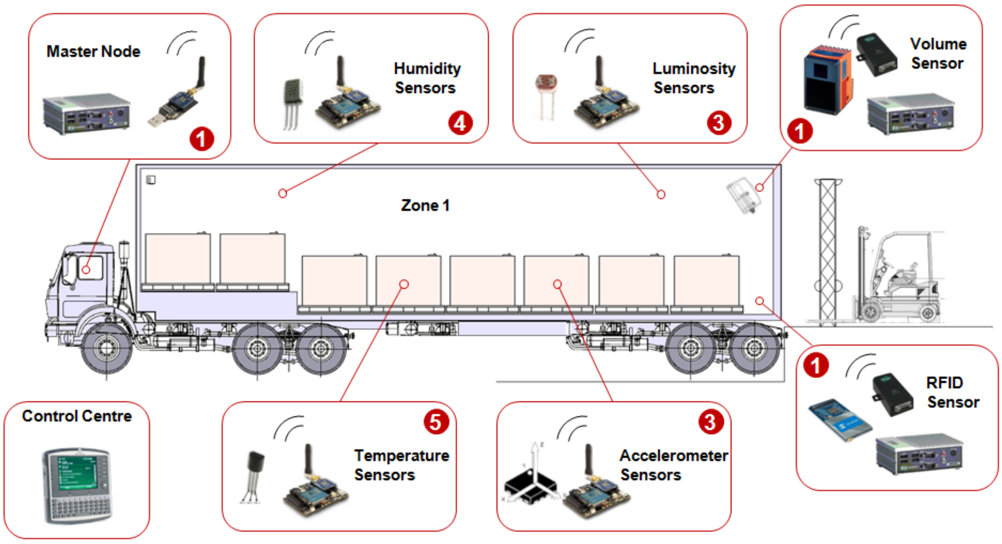

Figure 6.1: Scenario for perishable goods surveillance

6.1 Hardware description

In this particular application, the hardware of the sensor nodes consists of Waspmote boards battery powered with a ATmega1281 microprocessor and an XBee Pro radio module [20]. These sensor nodes are responsible for the acquisition of the environmental values: temperature (5 sensors), humidity (4 sensors), light intensity (3 sensors) and vibrations (3 sensors through 3-axis accelerometer). For RFID and volume sensor an embedded target with Intel Atom processor has been selected with an XBee Pro module adapter for radio communication.

The master node has been implemented over a PC based target with a Waspmote gateway. This platform includes other types of communications like Ethernet, Wi-Fi and USB, which enable messages exchange with the control centre or with other master nodes. The platform which acts as the control centre is a mobile computer Motorola VC6096 with Microsoft Windows Mobile 6.1 operating system.

As a result of the implemented plug-and-play protocol with node and service discovery, both an available service list from sensor nodes and their capabilities are automatically detected and stored by the master node when the sensor nodes join the network. At this point, the user is provided with all the information to request a high-level service through the control centre using the templates.

Two examples of possible high-level services requested by the user are described in the next sub-chapter.

6.2 Services composition examples

6.2.1 Example 1: "Temperature alarm when de load temperature exceeds 5 degress Celsius"

The parameters selected by the user for this service are:

- Sensor magnitude: Temperature
- Number of sensors needed : 4 along the container
- Spatial dispersion: along the container, not at a point.
- Event type: Alarm
- Threshold: 5ºC

The master node checks the feasibility of the service and then the automatic service composition starts with the selection of the sub-master node. In this particular case, the sensor node nearest to the master node is selected and responsible for the service. As mention above, the localization of this node is obtained through the RSSI (Received Signal Strength Indication) value between both the master node and the sensor node.

The equivalent parameters using the proposed formulation to the user requirements are shown in Table 6.1. The statistical function chosen is median value of all of measurements and the locID Zone 1 refers to inside the container. In this case, it is desired that the nodes chosen to form the virtual instrument are homogeneously distributed along the container, so the range selected for the spatial dispersion shows this situation. The behavior for this service is onAlarm.

After the sub-master selection and the sensors selection processes, the obtained result of the measurement service shows that only three temperature sensors satisfy the

Param	MS
FS	Temperature
N	4
ST	Median
locID	Zone 1
BH	onAlarm (T>5)
σ_s	[5,15]
Trigg	No

Table 6.1: Example 1 Temperature alarm

user requirements, therefore, the QoS index value is 0.75 (Equation 6.1). In this particular case, the virtual instrument is formed by an only measurement service. Equations 6.2 show these results. The value of the temperature is measured in Celsius degrees and the format for the time parameter is hh:mm:ss.

$$QoS_{MS1} = 1 - \frac{nr_{MS1} - ns_{MS1}}{nr_{MS1}} = 1 - \frac{4-3}{4} = 0.75 \tag{6.1}$$

$$\begin{aligned} MS &= \begin{pmatrix} V \\ t \\ QoS_{MS} \end{pmatrix} = \begin{pmatrix} 5.5 \\ 6:20:34 \\ 0.75 \end{pmatrix} \\ VS &= \begin{pmatrix} LV \\ t \\ LQoS_{MS} \end{pmatrix} = \begin{pmatrix} 5.5 \\ 6:20:34 \\ 0.75 \end{pmatrix} \end{aligned} \tag{6.2}$$

6.2.2 Example 2: "Conditions of the load during the transport: temperature, vibrations, luminosity and humidity"

The goal of this service is to obtain in a periodically manner the conditions of the goods under surveillance. For this purpose, the parameters selected by the user are:

- Sensor magnitudes: Temperature, vibration, luminosity and humidity
- Number of sensors needed: 4 temperature, 3 vibrations, 2 luminosity and 4 HR.
- Spatial dispersion: along the container, not at a point.

- Event type: Every 30 minutes

In this case, there are 4 measurement services corresponding to each one of the different family of sensors requested. As in the previous case, table 6.2 shows the translation of the user parameters into the proposed formulated parameters. The locID for all the homogeneous measurements is Zone 1 and the behaviour is onPeriod. The trigger signal for this case is the internal clock of the sensor node.

Param	**MS1**	**MS2**	**MS3**	**MS4**
FS	Temperature	Vibrations	Luminosity	Humidity
N	4	3	2	4
ST	Median	Max of period	Median	Median
locID	Zone 1	Zone 1	Zone 1	Zone 1
BH	onPeriod	onPeriod	onPeriod	onPeriod
σ_s	[5,15]	[5,15]	[5,15]	[5,15]
Trigg	Clock in node	Clock in node	Clock in node	Clock in node

Table 6.2: Example 2 Conditions of the load during the transport

As in the first example, the master node checks the feasibility of the service and then the automatic services composition initiates the selection of the sub-master node. The sensor node nearest to the master node is selected as responsible for the service. Once the sub-master is selected, it is computed the spatial dispersions of all the candidate sensors. Equation 6.3 shows that it is not possible to find all of the sensors required for each measurement service. The QoS indexes below one show this situation.

$$
\begin{aligned}
nr_{MS} &= [4, 3, 2, 4] \\
ns_{MS} &= [3, 3, 2, 2] \\
QoS_{MS} &= [0.75, 1, 1, 0, 5]
\end{aligned}
\tag{6.3}
$$

In this case, the association of the four measurement service forms the virtual instrument and a list of values is returned as the result of the virtual instrument service. As in the previous case the format of the timestamp is hh:mm:ss (Equation 6.4).

$$MS_1 = \begin{pmatrix} 25C \\ 6:20:34 \\ 0.75 \end{pmatrix} MS_2 = \begin{pmatrix} 0.9g \\ 6:20:34 \\ 1 \end{pmatrix}$$

$$MS_3 = \begin{pmatrix} 50lux \\ 6:20:34 \\ 1 \end{pmatrix} MS_4 = \begin{pmatrix} 60\% \\ 6:20:34 \\ 0.50 \end{pmatrix} \tag{6.4}$$

$$VS = \begin{pmatrix} LV \\ t \\ LQoS_{MS} \end{pmatrix} = \begin{pmatrix} \{25, 0.9, 50, 60\} \\ 6:30:34 \\ 0.81 \end{pmatrix}$$

CHAPTER 7

Conclusions and Future Work

In this Master thesis, a distributed measurement system capable of providing heterogeneous and configurable services to the user has been presented. The tasks involved in the development of this system have been the following ones:

- Applying SOA concepts for solving the restrictions of the wireless sensor networks. This approach offers an abstraction layer to manage the sensors as services.
- Designing a plug & play architecture for service and device discovery. With this mechanism all the information about the sensors and their main characteristics are available in the network.
- Developing an automatic service composition method. This method allows for the generation of the virtual instruments through the homogeneous sensors association. A novel formulation has been provided for this purpose.
- Adding a quality of service index to the measurement resulting from the user requested service. This index provides information about the fulfillment of the requisites of the virtual instrumentation service requested by the user.
- Developing the software modules needed for the implementation of this architecture. These modules allow for each device, executing their main tasks such as acquisition, communication and processing of the data as services.
- Applying the proposed method to the surveillance of perishable good with some examples of query services in order to demonstrate the validation of this measurement system.

As future work, a greater emphasis will be given to the services composition algorithms with more exhaustive tests of dynamic nodes (nodes switching on and off).

Another important challenge is the independence of the systems from a specific platform. Although the SOA approach solves part of this problem, an initial configuration of some parameters of the sensors is still necessary. The use of standards as TEDs (Transducer Electronic Data sheets) makes improvements possible in this sense.

The development of a semantic description of the high level services would be worth studying. The usability of the user interface would thus improve in a considerable way.

In the networking field, the use of the standard 6LowPan (IPv6 over Low power WPAN) over the 802.15.4 layers seems to be an investment for the future. Currently there are several research groups working in this direction.

Despite the existence of preliminary results of memory and battery consumption, it would be interesting to evaluate the overhead that this method adds to the power consumption and to the communications on deep. Studying the applicability of this architecture to other scenarios as train and plane cales or logistic centers would be interesting as well as future work.

Bibliography

[1] D. Angela, M. Ghenghea, I. Bogdan, *Supporting environmental surveillance by using wireless sensor networks*, Electrical and Electronics Engineering (ISEEE), 2010 3rd International Symposium on , vol., no., pp.216-219, 16-18 Sept. 2010 4

[2] R. Arce, *Nonlinear Signal Processing. A Statistical Approach*, Wiley:New Jersey, USA, 2005. 18

[3] T. Baugé, *Wireless Sensor Networks*, Thales Research And Technology (UK) White Paper, Sept. 2009 4

[4] M. Becker, B.-L. Wenning,C. Görg, R. Jedermann, A. Timm-Giel, *Logistic applications with Wireless Sensor Networks*. In: Proc. of HotEmNets 2010. 2010 4, 31

[5] J. Blumenthal, D. Timmermann, Resource-aware service architecture for mobile services in wireless sensor networks, in: Proceedings of the International Conference on Wireless and Mobile Communications (ICWMC), July 2006, pp. 34-39. 8

[6] F. Golatowski, J. Blumenthal, M. Handy, H. Haase, *Service oriented software architecture for sensor networks*, in: Proceedings of the International Workshop on Mobile Computing, 2003 7

[7] J. King, R. Bose, Y. Hen-I, S. Pickles, A. Helal, Atlas: a service oriented sensor platform: hardware and middleware to enable programmable pervasive spaces, in: Proceedings of the 31st IEEE Conference on Local Computer Networks, November 2006, pp. 630- 638. 8

[8] A. Kovacevic, J. Ansari, P. Mähönen, *Demo Abstract: Discovering Services in Mobile, Flexible and Heterogeneous Wireless Sensor Networks*, International Conference on Information Processing in Sensor Networks, 2009. IPSN 2009., San Francisco, CA, 13-16 April 2009

[9] M. Kushwaha, I. Amundson, X. Koutsoukos, S. Neema, J.Sztipanovits, OASiS: A Programming Framewor for Service-Oriented Sensor Networks. In Proceedings

of the 2nd IEEE/Create-Net/ICST International Conference on COMmunication System softWAre and MiddlewaRE (COMSWARE' 07), Bangalore, India, January 2007. IEEE Computer Society Press. 8

[10] J.P. Lynch, *An Overview of Wireless Structural Health Monitoring for Civil Structures*, Philosophical Transactions of the Royal Society of London: Series A (2007) 365, 345-372 4

[11] G. Mao, B Fidan, B.D.O. Anderson, *Wireless sensor networks localization techniques*, Computer Networks 51 (2007) 2529-2553 18

[12] P.S. Maybeck, *Stochastic Models, Estimation and Control Volume 1*, ISBN-10: 0124807011, June, 1979.

[13] E. Meshkova, J. Riihijarvi, F. Oldewurtel, C. Jardak, P. Mahonen, *Service-Oriented Design Methodology for Wireless Sensor Networks: A View through Case Studies*, IEEE International Conference on Sensor Networks, Ubiquitous and Trustworthy Computing, 2008. SUTC '08. , Taichung, 11-13 June 2008

[14] N. Y. Othman, R. H. Glitho, F. Khendek. The Design and Implementation of a Web Service Framework for Individual Nodes in Sinkless Wireless Sensor Networks, in Proceedings of the IEEE International Conference on Computers and Communications (ISCC'07), pages 941-947, Aveiro, Portugal, July 2007. IEEE Computer Society Press. 8

[15] C-Y. Shih, S.F. Jenks, *A Dynamic Cluster Formation Algorithm for Collaborative Information Processing in Wireless Sensor Networks*, 3rd International Conference on Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007, Melbourne, Qld. 3-6 Dec. 2007

[16] A. Rezgui,M. Eltoweissy, *Service-oriented sensor actuator networks: Promises, challenges, and the road ahead*, Computer Communications, 30:2627-2648, 2007. 7

[17] L. Ruiz-Garcia, P.Barreiro, J.I. Robla, *Wireless Sensor Networks Study for Real-Time Fruit Logistics Monitoring*, AgEng2008. International Conference on Agricultural Engineering & Industry Exhibition 31

[18] V. Vanitha, V. Palanisamy, N. Johnson, G. Aravindhbabu, *LiteOS based extended service oriented architecture for wireless sensor networks*, International Journal of Computer and Electrical Engineering, vol. 2, no. 3, 2010. 7

[19] Waspmote 802.15.4: Networking Guide, Document Version: v0.6 - 10/2011, Libelium Comunicaciones Distribuidas S.L. 29

[20] Waspmote, http://www.libelium.com/products/waspmote 32

[21] XBee®/XBee-PRO® RF Modules, Product Manual v1.xEx - 802.15.4 Protocol, Digi International Inc. 29

[22] X. Xu, M. Wu, C. Ding, B. Sun, J. Zhang , *Outdoor wireless healthcare monitoring system for hospital patients based on ZigBee*, Industrial Electronics and Applications (ICIEA), 2010 the 5th IEEE Conference on , vol., no., pp.549-554, 15-17 June 2010 4