SemSorGrid4Env

FP7-223913

WP 6. Fire Risk Monitoring and Warning in Castilla y León

D6.1 Requirements specification

Israel Liébana (Deimos Space)

Víctor Manuel Díaz (Deimos Space)

Agustín Izquierdo (Deimos Space)

Version 1.3

26/02/2009

Executive Summary

Traditional systems for preventing and detecting forest fires rely on methods that need too many human resources and are not automated. The proposal addressed in this document is to develop a system that will use cutting-edge technologies as Wireless Sensor Networks (WSN), Satellite Data and as well as historical data that will provide useful information to analyze and prevent fire risk situations.

The WSN, one of the main pillars, gets advance of its redundancy nature. A large amount of strategically deployed nodes with appropriate sensors can detect with high accuracy if a fire is set and where it is located. This, combined with the satellite data, will increase even more the accuracy, creating a system where the probability of preventing a fire will be some orders of magnitude higher than other methods.

The main added values of this proposal is the combination of both kinds of information in order to get a higher accuracy in the prevention and monitoring of fires and the in-network processing approach to minimize the data that comes out of the network for the maximization of the sensor battery life.

This document addresses the requirements needed to design and build an application for forest fire prevention combining information coming from real sensor networks deployed in one area of Castilla y León (Spain) and information of images from satellites. These requirements are derived from:

- The needs of different users like stakeholders and final users.
- The variety of phenomenon to be observed by the sensors.
- The characteristics of the images coming from the satellite.
- The basic functionalities required to be implemented by the architecture that supports and makes possible the rapid application development on top of it.

Note on Sources and Original Contributions

The SemSorGrid4Env consortium is an inter-disciplinary team, and in order to make deliverables self-contained and comprehensible to all partners, some deliverables thus necessarily include state-of-the-art surveys and associated critical assessment. Where there is no advantage to recreating such materials from first principles, partners follow standard scientific practice and occasionally make use of their own pre-existing intellectual property in such sections. In the interests of transparency, we here identify the main sources of such pre-existing materials in this deliverable:

- Section 5.2 contains material adapted from TinyOS web page http://www.tinyos.net
- Section 5.3 contains material from Crossbow Technology Inc www.xbow.com
- Section 5.4 contains material adapted from Contiki OS web page http://www.tinyos.net
- Section 5.5 and 5.6 contain material adapted from the following data sources
  - Crossbow Technlogy Inc www.xbow.com
  - Shock Fish SA http://www.tinynode.com/
  - BTNode http://www.btnode.ethz.ch
  - Contiki http://www.sics.se/contiki
  - Wikipedia http://www.wikipedia.org

Document Information

Contract Number	FP7-223913	**Acronym**	SemSorGrid4Env
Full title	SemSorGrid4Env: Semantic Sensor Grids for Rapid Application Development for Environmental Management		
Project URL	www.semsorgrid4env.eu		
Document URL			
EU Project officer	Daniel Quintart		

<table>
<tr><td>Deliverable</td><td>Number</td><td>D.6.1v1</td><td>Name</td><td colspan="4">Requirements specification</td></tr>
<tr><td>Task</td><td>Number</td><td>T.6.1v1</td><td>Name</td><td colspan="4">Identify the requirements posed by the developers, stakeholders and potentially also by the end-users of the test case.</td></tr>
<tr><td>Work package</td><td>Number</td><td>6</td><td></td><td colspan="4">Fire Risk Monitoring and Warning in Castilla y León</td></tr>
<tr><td>Date of delivery</td><td colspan="2">Contractual</td><td>To+6
28/02/2009</td><td>Actual</td><td colspan="3">28/02/2009</td></tr>
<tr><td>Code name</td><td colspan="3"></td><td>Status</td><td colspan="3">draft ☑ final ☐</td></tr>
<tr><td>Nature</td><td colspan="7">Prototype ☐ Report ☐ Specification ☐ Tool ☐ Other ☐</td></tr>
<tr><td>Distribution Type</td><td colspan="7">Public ☐ Restricted ☐ Consortium ☐</td></tr>
<tr><td>Authoring Partner</td><td colspan="7">Deimos Space</td></tr>
<tr><td>QA Partner</td><td colspan="7">National and Kapodistrian University of Athens (NKUA)</td></tr>
<tr><td rowspan="2">Contact Person</td><td colspan="3">Agustín Izquierdo</td><td colspan="4">Deimos Space</td></tr>
<tr><td>Email</td><td colspan="2">agustin.izquierdo@deimos-space.com</td><td>Phone</td><td>+34 918063450</td><td>Fax</td><td>+34 918063451</td></tr>
<tr><td>Abstract
(for dissemination)</td><td colspan="7">The proposal addressed in this document is to develop a system that will use cutting-edge technologies as Wireless Sensor Networks (WSN), Satellite Data and as well as historical data that will provide useful information to analyze and prevent fire risk situations.</td></tr>
<tr><td>Keywords</td><td colspan="7">Wireless sensor network, satellite images, fire prevention use case</td></tr>
<tr><td>Version log/Date</td><td colspan="3">Change</td><td colspan="4">Author</td></tr>
<tr><td>0.1</td><td colspan="3">Template</td><td colspan="4">A. Ibarrola</td></tr>
<tr><td>1.0</td><td colspan="3">First draft</td><td colspan="4">Deimos Space</td></tr>
<tr><td>1.1</td><td colspan="3">Revision of first draft</td><td colspan="4">Deimos Space</td></tr>
<tr><td>1.2</td><td colspan="3">Second revision, release for QA assessment</td><td colspan="4">Deimos Space</td></tr>
<tr><td>1.3</td><td colspan="3">Third revision after QA assessment of NKUA.</td><td colspan="4">Deimos Space</td></tr>
<tr><td></td><td colspan="3"></td><td colspan="4"></td></tr>
</table>

Project Information

This document is part of a research project funded by the IST Programme of the Commission of the European Communities as project number FP7-223913. The Beneficiaries in this project are:

Partner	Acronym	Contact
Universidad Politécnica de Madrid (Coordinator)	UPM	Prof. Dr. Asunción Gómez-Pérez Facultad de Informática Departamento de Inteligencia Artificial Campus de Montegancedo, sn Boadilla del Monte 28660 Spain #e asun@fi.upm.es #t +34-91 336-7439, #f +34-91 352-4819
The University of Manchester	UNIMAN	Prof. Carole Goble Department of Computer Science The University of Manchester Oxford Road Manchester, M13 9PL, United Kingdom #e carole@cs.man.ac.uk #t +44-161-275 61 95, #f +44-161-275 62 04
National and Kapodistrian University of Athens	NKUA	Prof. Manolis Koubarakis University Campus, Ilissia Athina GR-15784 Greece #@ koubarak@di.uoa.gr #t +30 210 7275213, #f +30 210 7275214
University of Southampton	SOTON	Prof. David De Roure University Road Southampton SO17 1BJ United Kingdom #@ dder@ecs.soton.ac.uk #t +44 23 80592418, #f +44 23 80595499
Deimos Space, S.L.	DMS	Mr. Agustín Izquierdo Ronda de Poniente 19, Edif. Fiteni VI, P 2, 2º Tres Cantos, Madrid – 28760 Spain #@agustin.izquierdo@deimos-space.com #t +34-91-8063450, #f +34-91-806-34-51
EMU Limited	EMU	Dr. Bruce Tomlinson Mill Court, The Sawmills, Durley number 1 Southampton, SO32 2EJ – United Kingdom #@ bruce.tomlinson@emulimited.com #t +44 1489 860050, #f +44 1489 860051
TechIdeas Asesores Tecnológicos, S.L.	TI	Mr. Jesús E. Gabaldón C/ Marie Curie 8-14 08042 Barcelona, Spain #@ jesus.gabaldon@techideas.es #t +34.93.291.77.27, #f ++34.93.291.76.00

Table of Contents

List of Images

List of tables

1. Introduction

This document contains the “Fire Use case Requirements Specification” version 1.2 for SSG4ENV’s work package 6, deliverable 6.1v1. It corresponds to the scenario specification of the fire use case which is focused on creating a fire risk monitoring and warning system in the Region of Castilla y León in Spain.

The system aims to prevent and detect fire combining two real-world real-time data sources, satellite data and measures taken from a deployed Wireless Sensor Network in the area. Both technologies follow different approaches. The satellite senses the environment from a long distance while the WSN does it locally, next or inside the monitoring area. Using the two approaches will be applied to demonstrate the state-of-the-art of technology and research inside the framework of SemsorGrid4Env new techniques and technologies to confront this environment risk.

1.1. Scope

This report is the result of task 6.1v1 of the project “Semantic Sensor Grid Rapid Application Development for Environmental Management (SSG4ENV)”.

The following list describes the work done for the present document regarding the requirements of the system and the contributions to the other work packages:

- Specification of the fire application’s requirements taking into account the needs of the stakeholders and the final users.
- First definition of the application graphical interface that aims to cover the basic needs of the different user profiles.
- Specification of the SemsorGrid4Env architecture’s requirements needed by the fire application basic functionality in order to be designed and built on top of the SemsorGrid4Env architecture.
- Identification of the main components of the SemsorGrid4Env architecture and their basic responsibilities.
- Specification of the sensor’s requirements constrained by the problem to be addressed, the technology to be used for in-network processing and the posterior development and validation phases of the embedded software.
- Market research about the available nodes and sensors useful for the fire application taking into account nodes in which in-network algorithms can run.
- Market research about the available development platforms for each node.
- Specification of the basic UML use cases derived from the fire application requirements.
- Specification of the UML sequence diagrams related to the UML use cases. The reason of this premature specification is the preliminary identification of the interfaces to be implemented by the architecture’s components.

1.2. Document Structure

The information within this document is presented as follows:

- Section 2: Technical Introduction. This section describes the scenario in which the sensor network will be deployed and the sorts of information that the application will cope with: real-time data measured by the WSN and interpreted data from images coming from satellite observation.
- Section 3: Fire Application’s Requirements: This section specifies the basic requirements of the application identified from the users, requirements regarding the technology of the sensor nodes and the data provided by the satellite.
- Section 4: Architecture’s Requirements: This section specifies the basic requirements of the architecture needed by the fire application case use.
- Section 5: Sensor Network’s Requirements: This section specifies the basic requirements that the sensors and the nodes or motes have to accomplish in order to cover the requirements identified in section 4 and the nature of the scenario described in section 3.
- Section 6: Application Interface. This section describes a first approach for the user interface of the application taking into consideration the different kind of users that potentially will use the application.
- Section 7: Fire Application’s Use Cases: This section specifies the basic use cases derived from the application’s requirements identified in section 4.
- Section 8: Fire Application’s Sequence Diagrams: This section specifies the sequence diagrams corresponding to the use cases specified in section 7.
- Section 9: Conclusion. This section resumes the content of this report, highlighting the justification for each section.

1.3. Applicable and Reference Documents

The following table lists a subset of Applicable Documents that have a direct impact on the content of this document.

Ref	Document Title	Document Reference	Version	Date
[AD1]	SSG4ENV Contract			
[AD2]	SSG4ENV Consortium Agreement			
[AD3]	SSG4ENV Technical and Financial, Management & Administrative Proposal			

Table 1. Applicable Documents

1.4. Acronyms

Acronym	Description
ADC	Analog to Digital Converter
CSV	Comma Separated Values

FPAR	Fraction of Photosynthetically Active Radiation
HR	High resolution image (pixel size of 5-10 meters)
IP	Internet Protocol
LAI	Leaf Area Index
LST	Land Surface Temperature
NDVI	Normalized Difference Vegetation Index
MR	Medium resolution image (pixel size of 15-30 meters)
OS	Operating System
RF	Radio Frequency
SQL	Structured Query Language
SSG4ENV	Semantic Semsor Grids Rapid Application Development for Environmental Management
TCP	Transmission Control Protocol
UML	Unified Modeling Language
VNC	Virtual Network Computing
WSN	Wireless Sensor Network
XML	Extended Mark-up Language

Table 2. List of acronyms.

1.5. Definitions

A lysimeter is a device used to measure the rate of evapo-transpiration from a small (typical 0.25-4 m2) patch of vegetation.

An hygrometer is a device used to measure humidity

A dendrometer measures the volume and height of individual trees

A porometer a is an instrument that measures the diameter of a pore at its most constricted part, the largest pore diameter, the mean pore diameter, and the pore distribution in a porous material.

A solarimeter is a pyranometer used to measure combined direct and diffuse solar radiation

A ceptometer is an array of photodiodes sensitive to Photo synthetically Active Radiation. It measures the distribution of sun flecks (direct solar radiation which has passed through the canopy un-attenuated) as a proportion of the leaf area index.

2. Technical introduction

In the following chapters, some technology related aspects like sensor technology, node platforms and prototypes, satellite data and software layers will be analyzed.

On the other hand, the real added value of the fire application is the combination of the real-time measurements provided by the sensor network and the interpreted data coming from the satellite images. For this reason, a chapter is devoted to the correlation between both sorts of data.

2.1. Sensors

The network nodes will have a group of sensors connected in order to help the system for fire detection as well as its prediction.

All the nodes will not have to sense the same parameters as in some cases the variation of the same kind of measurement in a close area will not provide extra information. On the contrary, some other measurements must be retrieved in each node to detect whether or not in its surrounding an event is detected.

Some of the parameters that the appropriate sensors must measure are listed in the following list:

Variable	Image-processing	On land sensors	Purpose
Soil temperature	Heat, LST (LR)	Soil temperature / temperature in plant	Fire Danger / Emergency
Soil moisture	NDVI, LST (MR)	Hygrometer (Soil), lysimeter	Fire Danger / Forest Management
Humidity dead matter	Fuel alive (MR)	Hygrometer (Soil)	Fire Danger / Forest Management
Air humidity	Band of water vapour (LR)	Hygrometer	Fire Danger
Vegetative growth	fPAR, NDVI, LAI, biomass, etc.).	Flow of sap, dendrometers	Forest management / Inventory taking
Photosynthetic activity	fPAR (MR, HR and original bands)	Flow of sap, porometer	Forest management / Inventory taking
LAI	LAI (MR, HR and original bands)	Ceptometer / Solarimeter	Forest management / Inventory taking
Evapotranspiration	NDVI, LAI, biomass, and so on.	Lysimeter, porometer, weather variables, and so on.	Forest Management
Changes in vegetation cover (demolition, arson, mortalities, illegal short, etc.).	And changes in Anomaly Detection: Optical sensors (MR, HR and original bands)	Surveillance cameras	Inventory taking / Forest Management
Fraction accommodate Cover (FCC)	Optical (HR and original bands)	Light sensor under canopy trees	Inventory taking/ Forest Management
Quality Station	Optical (MR, HR and original bands) fPAR, NDVI, biomass, and so	Dendrometer/ Hygrometer,	Forest Management

	on.		
Solar radiation incident	Irradiance (MR, HR and original bands)	Solar radiation incident	Forest Management
Water stress	NDVI, fPAR, and so on.	Hygrometers branch / Weather	Forest Management

Table 3. Proposed list of sensors.

For the first deployment, that will be the first approach to a working system, a more common set of sensors which can be easily obtained in the market could be:

- Temperature sensors.
- Relative humidity sensors.
- Soil moisture sensors.
- Leaf moisture sensors.
- Smoke sensors.
- Wind speed / direction sensors.

2.2. Coverage

The system is going to mix data provided from both satellite images and on-site sensors, so by the time of considering how to distribute the sensors in the selected physical environment, the satellite resolution has to be kept in mind. At least a sensor should be located in each minimum coverage area (pixel) provided by the satellite. Sometimes this area will be too large for the node radio range, so the node density might be increased.

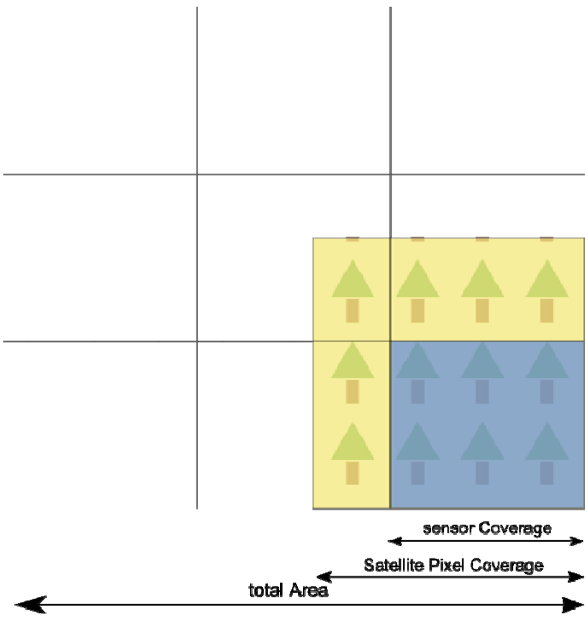

Figure 1: Sensor coverage versus image resolution.

2.3. Network topology

All the nodes will be connected to the network using the same radio interface for convenience. The topology should be as flexible as possible, relying on the network stack to build up a dynamic and self-configuring mesh network that will provide some guarantee of a usable network. It means, independently a part of the network would have any kind of problem; the rest of the network would work seamlessly. Those problems can be originated by:

- Battery depletion
- Node failure
- Obstacle which could reduce the radio signal among nodes

Nevertheless the physical topology must be known by the system, including the unique ID of the nodes and their position in the space, since it will work with data provided by the satellite.

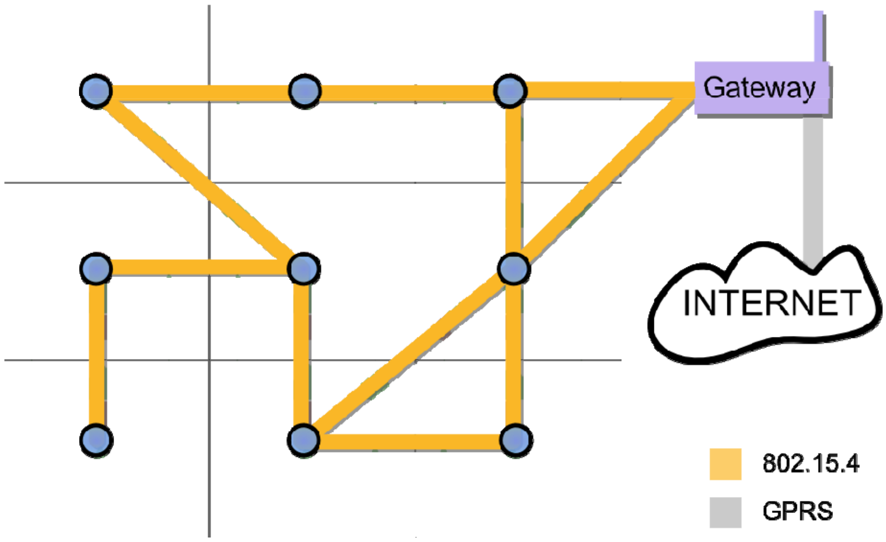

Figure 2: Sensor mesh and external communications link.

The nodes will transmit the sensor data using multi-hop routing instead of just 1 hop to reach the gateway. This solution will provide some advantages like:

- Low-cost network. The hardware would not be too complex.
- Increase of battery life due to a lower power is needed for sending data to its neighbor.
- A high-density network for better sensing.
- Tolerate random network faults as the data packets can be delivered using different paths.

The gateway is the window to the world; it will be a special node that can be as well the coordinator for the network. It should be permanently powered and a different network interface such a GPRS module connected to an IP network. Moreover the processing unit can be more powerful to perform different tasks like data processing, data aggregation or act as a data provider.

2.4. WSN measures and Satellite Images correlation

The two main data sources involved in the project are the satellite images and sensor measures from the WSN. Independently, both of the data sources can prevent and detect fires with some degree of exactitude.

In the present scenario both data sources, satellite imagery and the measures provided by the sensor network, will be used all together trying to cover the lack of functionality of each technology.

Satellite imagery is a very good option when covering a large area in some electromagnetic ranges. There are some cases where the satellite cannot provide or is not viable due to economic reasons, a sufficient and detailed data about fire detection. In that case, the WSN due to its relatively cheap hardware and deployment can complement the system with more accurate and local data than the satellite in many areas. As well satellite imagery it is going to be very useful in the system as it can analyze a large number of parameters in a large area with the possibility of analyzing areas which surrounds the region of interest for possible data extrapolation.

Each of the nodes that forms the WSN must be geo registered with GPS coordinates in order to correlate data with the satellite imagery, once both data sources will be adjusted to work in the same area, the previously grid organization commented and showed in **Figure 3** and **Figure 4**, will be used.

Using this approach, the satellite images can be sliced to fit a minimum of a pixel in the area covered by each node. As many pixels that can fit in each of the squares that forms the grid, the more accurate the system will get.

Both of the data sources will be given as two big squared areas divided into tiles, those tiles will have a correspondence 1:1 although sub tiles in the satellite data is allowed. These tiles have to match in position and size, so a tile in the satellite image has to have the same position in the space and size as the tile in the grid formed by the WSN.

In some cases where the node is offline, this approach cannot be followed as it is. As fist approach if the data of some position of the grid is missing, this will be interpolated using adjacent cells.

3. Fire application's requirements

3.1. Purpose

This task is related to the requirement specification of the fire use case application derived from the needs of the stakeholders and final users and the sorts of information to be used by the applications. This task also characterizes the different profiles of users that will use the application.

3.2. Specification

The following table contains the identified high-level requirements the fire use case application should fulfill.

Requirement	Description
Req-010	The application will display the geographical area in which the sensors were deployed.
Req-020	The application manages and represents graphically the different kind of data organized in layers.
Req-030	The application has the following layers: some layers for the representation of the sensor network, some layers for the representation of the data coming from satellite images, some layers for the representation of the combined data (sensor network and satellite images) containing the fire risk index and some layers for the representation of the historical data.
Req-040	The application will geo-locate every deployed sensor over the geographical area in a layer. This layer describes each sensor showing its geo-position, its current state and its current measured observations.
Req-050	The application will show the data coming from the image satellite in a layer.
Req-070	The application will display the combined information composed of the observations coming from the sensor network and the data interpreted from satellite images in a certain layer.
Req-080	The application will display the historical data measured by the sensor network in a certain layer.
Req-090	The application will display the historical data interpreted from the satellite images in a certain layer.
Req-100	The application will display the combined historical data (the data from the network and the data from the satellite images) in a layer.

Req-110	The user will be able to choose the layer(s) to be displayed in order to ease the monitoring through the application.
Req-120	The application will work in two modes: the surveillance mode and the manual mode. Both modes can be configured by the user.
Req-130	In surveillance mode, the application will display the information provided by the network and the satellite without user intervention. The application receives the information in asynchronous way. The refreshing information frequency is dependent of the new and useful information from the point of view of the application.
Req-140	In manual mode, the user could obtain the information provided by the network and the satellite in real time.
Req-150	In both surveillance and manual working modes, the application will display the information according to the selected layers.
Req-160	The application will cope with different sort of fire risk levels. Each fire risk level will be defined taking into account threshold values for each measurement provided by the sensor and the data interpreted from the satellite images.
Req-170	The application will allow the user to set the current fire risk level. This selected level affects the new displayed information in the different layers.
Req-180	In both working modes, the user may select an observation sub-area for monitoring. After that, the application will only display data from this sub-area. This operation affects to the information received from the network and from the satellite images in the sense that it is only referred to the configured sub-area. This operation lets the user to optimize the communication between the application and the network and the repository containing the satellite images, so the monitoring task could be better performed.
Req-190	In manual mode, the user could look up measurements observed by the sensors during a past temporal interval and contained in the historical database. This kind of information will be displayed into the associated layer.
Req-190	In the manual mode, the user could look up historical data provided by the satellite images. This kind of information will be displayed into the associated layer.
Req-200	In the manual mode, the user could look for historical combined information: historical data observed by the sensors combined with satellite historical data. This kind of information will be displayed into the associated layer.

Req-210	The user will be able to watch the sensors of the network grouped by similar measurement values. This operations lets the user to view and control different geographical zones in order to discover potential paths where a potential fire could spread. This kind of information will be displayed into the associated layer.
Req-220	The application will display the working state of each sensor in the network every moment. This kind of information will be displayed into the associated layer.
Req-230	The resolution of the satellite images is 32 m. by 32 m, so a pixel in the image represents approximately this area.
Req-240	The topology of the network, the sensor coverage and the image resolution has to be defined taking into account their interdependencies.
Req-280	The required physical sensors have to be compatible with the chosen sensor nodes.

Table 4. Requirements of the application.

3.3. User profiles

The application allows different user profiles interact with the system actively or passively depending on the access granted.

Among all the different users who could potentially use the application, three of them are characterized and described below.

- Non-registered visitor: This profile is focused on the occasional user who wants to get information about the project itself and as well as to take an overview of how the system works. Some public node’s information as location and some data, measured with the sensors, can be seen by the user.
- Operator: This profile is focused on the people who will be in charge of operating directly with the system. The main role of the user is to supervise the status of the system and act according the situation. The operator should be able to view all the individual measures taken by the nodes and as well the computed measures and alarms, previously configured by this user type, that the system possible would trigger.
- Advanced user: This profile is intended to be used by researchers and members of the project who want to control in some degree the system and monitor all the parameters which can be involved in its research. For the mentioned reason, extra information like network meta-information, raw data outputs, etc should be provided.

As the application will evolve and new requirements will establish, new characterization of users will appear. Mainly, those concerned to research, as different fields need different requirements.

4. Architecture's Requirements

4.1. Purpose

This task is related to the requirement specification of the SemsorGrid4Env architecture identified and needed by the fire use case application.

4.2. Specification

The approach taken to specify and design the SemsorGrid4Env architecture is a top-down specification approach. The first step in this approach is to assure a complete coverage by the architecture functionalities to the use cases' requirements. A second step will be to generalize these basic functionalities of the architecture to make possible the rapid development of other related applications.

The following table contains the identified high-level requirements of the SemsorGrid4Env architecture.

Requirement	Description
Req-400	The architecture should have a service for registering foreign sensor networks allowing them to associate meta-information to the service offered by network. The register should consider at least the following service features: geographic area covered by the network, measures offered by the sensors, current state of the sensors and quality of service indexes.
Req-410	The architecture must be able to provide the meta-information related to each sensor of the network. The meta-information should contain at least the following information: the geo-location of each sensor, the measurement and its working status.
Req-420	The working status of each node contained in the meta-information has to be notified by the architecture to the application every time this status changes.
Req-430	The information provided by the architecture is retrieved by running a defined query. This mechanism affects to the information measured by the sensor network, to the information coming from satellite images and to the historical information.
Req-440	The architecture maintains a catalogue of queries defined by the application over data services as: sensor networks, satellite images or historical data.
Req-450	The architecture will provide an asynchronous data communication mechanism, probably based in a subscriber (application) and publisher

	(system) mechanism.
Req-460	The application will be able to set a defined query as the active query. After that, all operational information received by the application is retrieved by this query.
Req-470	There is only one active query associated to a data service (network sensor or satellite images) and a certain application at the same time.
Req-480	The architecture should provide the subscription to a certain type of event in order to make possible the final application to receive information asynchronously.
Req-490	Each event managed by the architecture is associated with a query defined by the application containing the retrieval method for obtaining the desired information.
Req-500	The architecture must allow synchronized data communication, based on a request-answer mechanism.

Table 5. SemsorGrid4Env architecture's requirements.

4.3. Interactions with the architecture's components

The network topology will follow a Mesh configuration, so potentially any node could establish a wireless connection to any other node if the radio signal is appropriate. All the nodes will be distributed along the field trying to follow a grid pattern.

This configuration matches the requirements of the satellite data mash-up due to the correlation of 1 image pixel to 1 one sensor, which is the approach to be followed. Another reason for a grid configuration is the easiness of data interpolation among sensed measures and node addressability.

Concerning the addressability, every node will be numbered according to its position on the grid, as showed in the following representation.

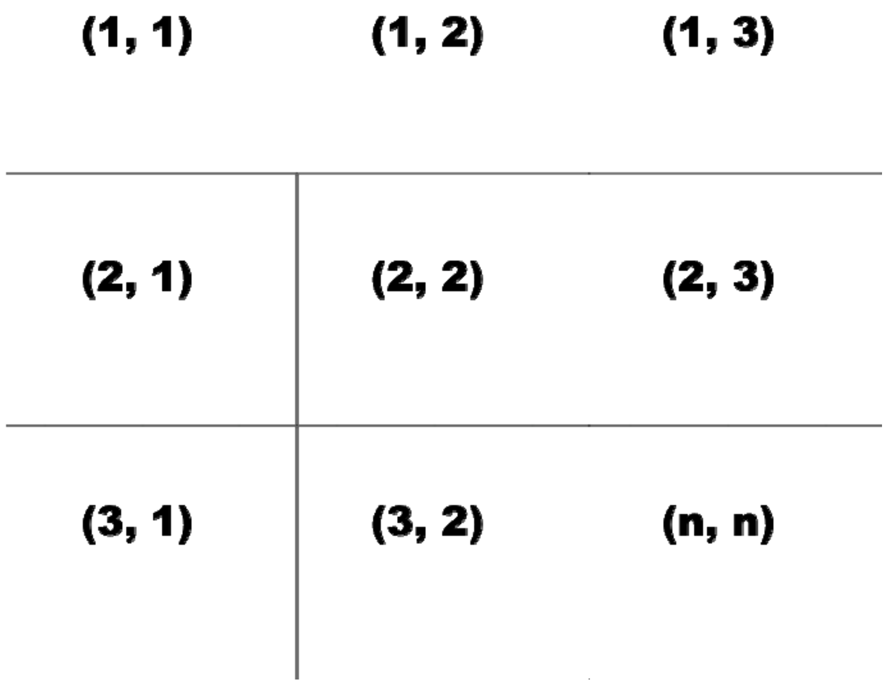

Figure 3: Node addressability

This grid representation is stored in the Registry component of the architecture. Every node in the network can be addressed using three methods, grid address and the GPS position and normal addressing method based on the network node identifier.

Once every node is geo-located, the Registry can return, when queried, the region occupied by the whole network determining the boundaries the network defines.

Another important factor is identifying all the sensors each node has from the perspective of building a network. This network will not be physical defined, it will be an *overlay or virtual network* based on sensor information, being useful when working solely with sensor data abstracting the network layer. In this way, we could separate the system into layers depending on the sensors used.

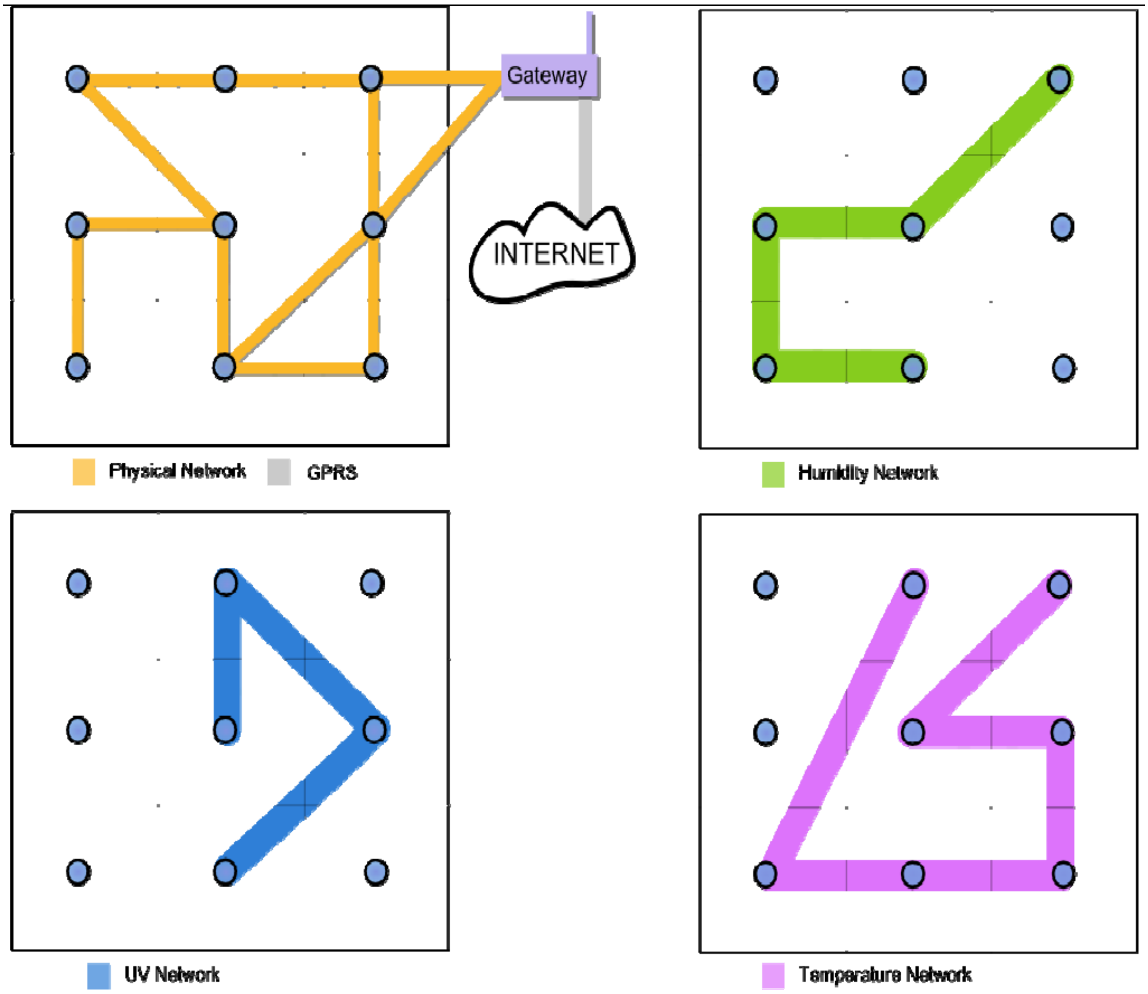

Figure 4: Sensor based Overlay-Networks

To sum up, the Registry needs to have a mechanism to classify the registered networks depending on the sensing variables, the geo-position of each node, the covered area and the state of the network and nodes. All this information must be provided by the WSN network once registered, as well as some periodic updates, so an application can retrieve the information of the network using the Registry by position in the space, position on the virtual grid and used sensors.

The following table contains an overview of different methods to be implemented by the Registry as well as useful operators and functions.

Network queries

getNetworkTopology(network_id)	Gets a structure with the actual network topology
getNodeNeighbours(node_id)	Established links between neighbor nodes of a nodeId
getQualityLink(node_id, node_id)*	Wireless Link quality in dBm among two nodes n1, n2

getBatteryLevel(node_id)*	Battery (voltage) level of a node_id
getPhysicalDistance(node_id, node_id)	Gets the distance in meters from two nodes, interesting to compare link quality and distance.
getNearestNode(Longitude, Latitude)	Gets the id of the nearest node from a point in space, useful to correlate the sensor data with the satellite one.
getNodeFromGrid(Xposition, Yposition)	Gets the node corresponding to the grid number

Table 6 Network Queries

Sensor queries

getNodes()	Retrieves all the nodes in registered in the system
getActiveNodes()	Get only the nodes which are turned on
getSensors(node_id)	Get a list of working sensors in a node
getLastMeasure(Array node_id, Array sensor_id)	Get lasts measures for the given nodes and sensors
getHistoryMeasures(node_id, sensor_id, time period)	Retrieves the history of a node and a sensor in a defined time interval
getHighestMeasure(sensor_id)	Gets the highest measure for a certain sensor in the network
getCommitFrequency(node_id)	Gets the frequency of the data sent from a sensor to the middleware. This data can be used to detect problems in the network as node overuse.

Table 7 Sensor Queries

Operators

MAX	Maximum value
MIN	Minimum value
AVG	Average value of a given set
dValue/dT	Differential change of a measured value along the time.

Table 8 Operators

* Some of the queries could not be done as it depends on the functionality provided in the hardware layer

5. Sensor Network's Requirements

This section presents a market research for the different nodes and Operating Systems for WSN available in the market that fit the fire prevention use-case within the SSG4ENV project. The main constraint in this search is the need for the utilization of in-network processing algorithms in order to minimize the amount of data coming out of the network as to extend the life of the sensor's batteries too.

The aim of this document is to guide the WP6 collaborators, UNIMAN and NKUA, which are responsible for the running of these algorithms, towards a proper election of the nodes to be used in the demonstration of the aforementioned technology.

5.1. WSN Platforms

This chapter describes the most widely spread OS and software layers for nodes and meshes within a WSN.

5.2. TinyOS

TinyOS is an embedded operating system used in Wireless Sensor Networks nodes. It is written in a dialect of the C language called nesC. Its main features are:

- Small kernel footprint.
- Architecture based in components.
- Well separated abstraction layers. Clearly separated in each interface.
- It is pretty good adapted for its usage in resource-constrained motes in energy, processing power and storage and bandwidth.
- Every operation is divided in phases.
- Event-driven operation system.
- Concurrency.
- De-facto industry standard.

5.3. MoteWorks

MoteWorks is a set of software and layers across the whole WSN that present a complete solution for deploying a sensor network. MoteWorks provides with:

- The mote tier, where XMesh resides, is the software that runs on the cloud of sensor nodes forming a mesh network. The XMesh software provides the networking algorithms required to form a reliable communication backbone that connects all the nodes within the mesh cloud to the server.
- The server tier is an always-on facility that handles translation and buffering of data coming from the wireless network and provides the bridge between the wireless

Motes and the internet clients. XServe is the primary server tier application, and can run on PC or Gateway.

- The client tier provides the user visualization software and graphical interface for managing the network.

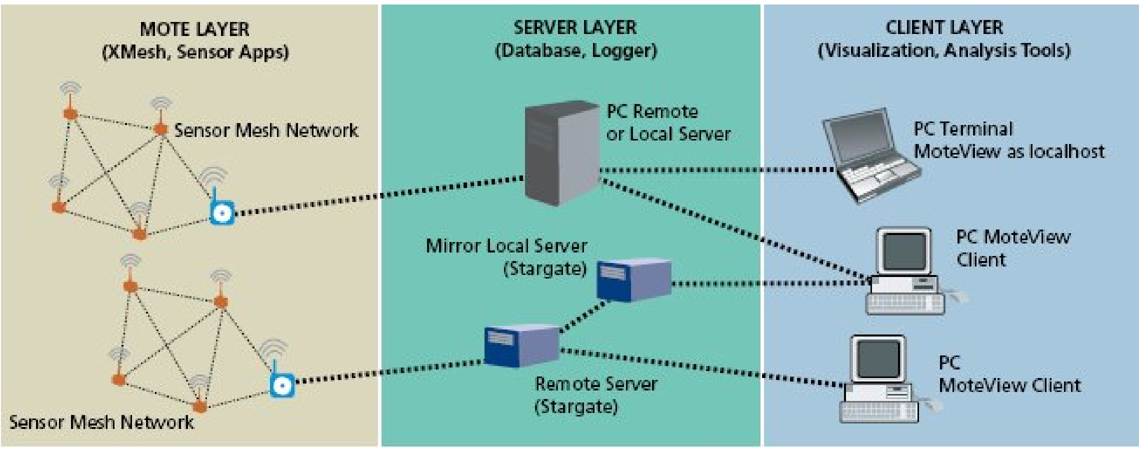

Figure 5: Architecture layers of MoteWorks

MoteWorks supports **only Crossbow MICA and IRIS series** Mote hardware and sensor boards: IRIS, MICAz, MICA2 and MICA2DOT and **it works only with TinyOS**. Some components of TinyOS are replaced by its analogous in MoteWorks as XMesh, XOtap and XRadio.

The modules that compose MoteWorks are detailed in the following paragraphs.

5.3.1. XMesh

The main features of XMesh are:

It provides dual support for the ZigBee standard and advanced mesh networking.

- It includes support for self-forming, self-healing networks and it is optimized for dynamic routing.
- XMesh automatically determines the optimal route through the network to the gateway, minimizing the power requirements throughout the network. When an RF link degrades in quality or becomes unavailable, XMesh automatically re-routes messages through other nodes in the network.
- XMesh fully supports bi-directional message communication throughout the network with fast alert/alarm propagation to support safety and security applications with stringent, response time requirements.
- Multiple gateways / sinks are not supported

5.3.2. XOtap - Over-the-air-programming & Remote Update

Over-the-air-programming and updating of sensor nodes is provided through MoteWorks' XOtap capability. This over-the-air-programming system allows motes to be programmed reliably, peer-to-peer or via multicast over a low data rate wireless network.

5.3.3. XRadio

XRadio provides low power radio protocol for battery-powered devices

5.3.4. XServe - Gateway Server Middleware

XServe is a software component that acts as middleware for the sensor network.

It can log all the incoming data as CSV file format or Database.

The data can be accessed using:

- XML interfaces
- SQL queries when raw data is required
- Locally or through a TCP/IP network

5.3.5. XSniffer

XSniffer displays radio communication for protocol debugging.

5.3.6. MoteConfig

MoteConfig is a simple graphical interface to program nodes flash memory and configure firmware images.

5.3.7. MoteView - Gateway Server Middleware

MoteView provides visualization and analysis of multiple sensor data streams. MoteView provides with:

- Pre-compiled firmware for simple deployment of periodic sensing applications.
- Unit conversions: it seamlessly converts raw ADC sensor values to final engineering units.
- Charting: it graphs sensor data over time with zoom/pan tools for detailed analysis.

- Topology view of the sensor network.
- Tabular display of sensor data across all nodes.
- Scroll backward and forward in time for a playback of sensor data.
- Health: quick view node outages, and detailed link quality information.
- Export: it manages database with import / export from SQL to CSV formats.
- Commands: it provides with a simple interface to control and actuate individual nodes.
- Alerts: it sends an email or text page when a sensor value crosses a critical threshold.

There are two licenses available for MoteWorks:

- MoteWorks Standard Edition. It is available via free download. Allows redistribution and use in source and binary forms under most circumstances involving academic, research, personal, evaluation, or non-profit use.
- MoteWorks Enterprise Edition. Includes full source to XMesh, XServe and CVS access. Additionally, customers with an Enterprise license will get priority support with a 24 hour response time. Standard support is via e-mail only and does not offer a guaranteed response time. The enterprise license allows commercial or for-profit use on Crossbow branded hardware.

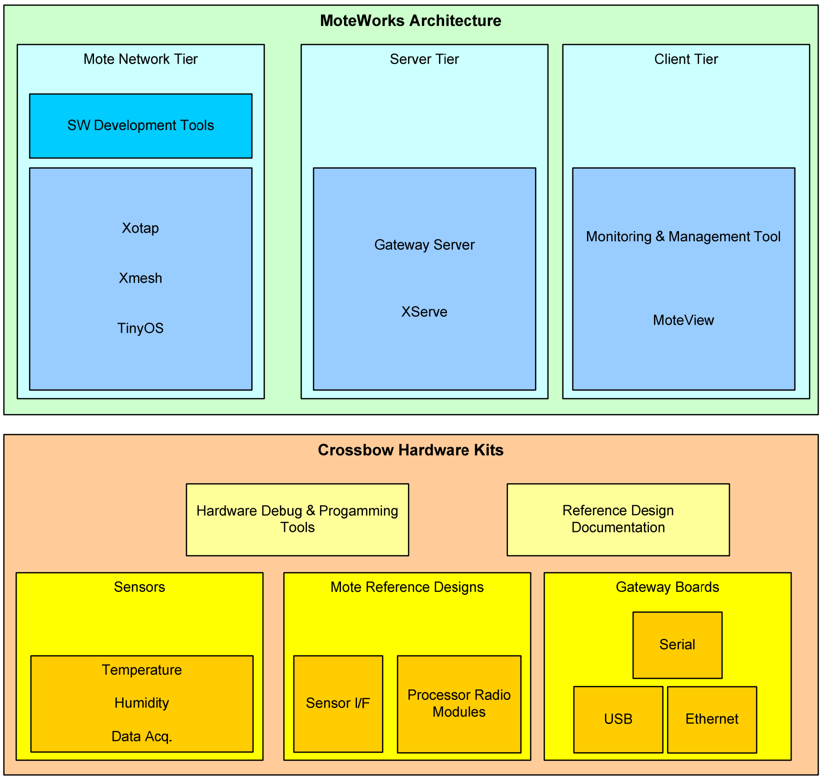

Figure 6: Components of MoteWorks.

5.4. Contiki

Some important functionalities of Contiki are:

- uIPuIP- lightweight TCP/IP stack and IPv6 support
- Multi-tasking
- Over the air reprogramming a single component of application or the entire firmware, taking around 30 seconds for reprogramming one node and approximately 30 minutes in 40 minutes network.
- It supports many processors in the market
- Event-based concurrency model
- Event-driven systems
- Pre-emptive multi-threading
- Lightweight proto-threads
- Dynamically-loadable programs and services
- Most programs run directly on top of the kernel
- Power save mode
- VNC support

5.5. *Software platforms comparison*

The table below shows a comparison of the main software platforms to be considered in the framework of the SSG4ENV project.

	TinyOS	MoteWorks	Contiki
Multiple Gateways / Sinks per cluster	Yes	No (It seems is a XServe limitation)	Yes
Code Size	Lower than Contiki	Similar to TinyOS	Higher than TinyOS
Over the air reprogramming	Yes, Deluge module	Yes, XOTap	Yes
Dynamic modules loading (for programming)	Yes	No	No
Open Source development	Yes, BSD License	Source code is provided in Enterprise Edition but it seems not all the code is open source.	Yes, BSD License
Compiler	nesC and architecture dependent compiler	nesC and architecture dependent compiler	architecture dependent compiler
Simulator	Tossim	-	Cooja

	http://docs.tinyos.net/index.php/TOSSIM		http://www.sics.se/~fros/cooja.php
Power profiling	PowerTossim (simulator) http://www.eecs.harvard.edu/~shnayder/ptossim	-	Contiki 2.1 or higher http://www.sics.se/contiki/current-events/contiki-2.1-released-with-new-energy-profiling-features.html
	Avrora (hardware platform dependent, Atmel AVR) http://compilers.cs.ucla.edu/avrora		
Development tools	Cygwin for non-UNIX environments is required http://www.cygwin.com TOSdev for TinyOS 1 http://selab.csuohio.edu/dsnrg/tosdev/index.html TinyOS 2 Eclipse plug-in http://tos-ide.ethz.ch/wiki/index.php	Same as TinyOS MoteConfig XSniffer Programmers notepad	Cygwin for non-UNIX environments is required http://www.cygwin.com All the development tools are included in Contiki's package and in a LiveCD (Instant Contiki) http://www.sics.se/contiki/current-events/the-instant-contiki-development-environment-1.0a.html
self-formation network	Not by default	Yes, XMesh	Not by default

self-healing network	Not by default	Yes, Xmesh	Not by default
Network and nodes visualization	Oscilloscope – displays sensor readings http://docs.tinyos.net/index.php/Sensing#The_Oscilloscope_application TinyViz	MoteView	-

Table 9. Software platforms comparison.

As a first review of the three options, the following advantages of using MoteWorks over a standalone TinyOS or Contiki have been observed:

- Better out-of-the-box solution.
- Complete set of development and forensic tools.
- Integration with the XServe middleware and the environment provided for its development.

The main disadvantages are:

- TinyOS 1.1 based (released in December 2005) when the latest release is TinyOS 2.1.0 (August 2008). TinyOS v1.1 does not seem to be a good choice for the project. On the other hand, TinyOS v2 could be a good alternative (better performance and general behaviour).
- MoteWorks only works with Crossbow nodes. Hardware catalogue will be reduced.

It is important to check the following points:

- How MoteWorks license will play in the development of the research and the differences with the enterprise version.
- How truly open it is for source code modification.
- To compare MoteWorks with actual versions of TinyOS and Contiki. Documentation is a bit unmaintained so collaboration would be appreciated.

5.6. *Hardware Platforms*

The table below provides a comparison between different nodes of the WSN that may fit for the SSG4ENV fire prevention demo use-case.

	Node Name	Microcontroller	Transceiver	Program Space	Ram	External Memory	Integrated Sensors	OS	Approx. price
	BTnode	Atmel ATmega 128L (8 MHz @ 8 MIPS)	Bluetooth (2.4 GHz) Chipcon CC1000 (433-915 MHz)	128 Kbyes	4 KBytes	64 + 180 KBytes	--	BTNut TinyOS Contiki in progress	200€
	IMote 2 (2006)	Intel PXA271 XScale® Processor at 13 – 416MHz	Bluetooth ChipCon CC2420 802.15.4 30 m without antenna	256K SRAM	32MB	32MB	--	TinyOS Linux SOS	299$

	Node Name	Microcontroller	Transceiver	Program Space	Ram	External Memory	Integrated Sensors	OS	Approx. price
	IRIS	Atmel ATmega1281	2.4 GHz IEEE 802.15.4 300 m	128K	8K	512K bytes	--	TinyOS MoteWorks supported Contiki	115$
	KMote	TI MSP430 microcontroller	250 kbit/s 2.4 GHz IEEE 802.15.4 Chipcon Wireless Transceiver	48KBytes	10Kbytes		Temperature, Light, Humidity	TinyOS SOS	80€
	Mica2	Atmel ATMEGA103 4 MHz 8-bit CPU	RFM TR1000 radio 50 kbit/s	128 KBytes	4KBytes	1MB Flash	--	TinyOS MoteWorks supported	--

	Node Name	Microcontroller	Transceiver	Program Space	Ram	External Memory	Integrated Sensors	OS	Approx. price
	MicaZ	ATMEGA 128L	Chipcon 868/916 MHz 250 kbps 75-100m	128K RAM	4Kbytes	512Kbytes	--	TinyOS MoteWorks supported Contiki in progress	99$
Not available	Rene	ATMEGA 128	802.15.4/ZigBee compliant RF transceiver	4K RAM	--	128K Flash	--	TinyOS MoteWorks supported	--

	Node Name	Microcontroller	Transceiver	Program Space	Ram	External Memory	Integrated Sensors	OS	Approx. price
	TelosB	ATMEL8535	Chipcon CC2420 2.4 GHz 802.15.4	48Kbytes	10KBytes	1 MB	Temperature Light Humidity	Contiki, TinyOS, SOS MantisOS	129$

	Node Name	Microcontroller	Transceiver	Program Space	Ram	External Memory	Integrated Sensors	OS	Approx. price
	TinyNode184	16MHz Texas Instruments MSP430 microcontroller (MSP430F2417)	868 / 915MHz Semtech SX1211 ultra-low power wireless transceiver (3mA in receive mode, 25mA in transmit mode at +10dBm) 200 m	92 KBytes	8 KBytes	512KBytes	--	TinyOS Cotiki (unofficial)	73 €

	Node Name	Microcontroller	Transceiver	Program Space	Ram	External Memory	Integrated Sensors	OS	Approx. price
	TinyNode584	16MHz Texas Instruments MSP430 microcontroller (MSP430F2417) with integrated temperature sensor	868 / 915MHz Semtech SX1211 ultra-low power wireless transceiver 25 KBps Bit rates up to 200kbit/s, NRZ coding 76 KBps 500 m	48KBytes	10KBytes	512K bytes	--	TinyOS Cotiki (unofficial)	91 €

	Node Name	Microcontroller	Transceiver	Program Space	Ram	External Memory	Integrated Sensors	OS	Approx. price
	T-Mote Sky	Texas Instruments MSP430 microcontroller	250 kbit/s 2.4 GHz IEEE 802.15.4 Chipcon Wireless Transceiver	48KBytes	10 Kbytes	1 MB	--	TinyOS Contiki	--

Table 10. Nodes comparison.

Most of the reviewed nodes do not provide any kind of built-in sensor. Due to its miniaturization, direct connection of any kind of sensors seems difficult and an external board with sensors and as well as analog and digital I/O must be used.

5.6.1. Data memory constraints

Most of the listed platforms above have a small amount of data storage. On the contrary only the iMote2, which provides as well a big amount of processing power, provides great capabilities of data storage.

An option for upgrading the data memory should be done although for some providers it is not possible.

5.6.2. Conclusions

It has been found that related to the software layer, Operating System, Middleware and programming easiness, the integrated solution provided from Crossbow is the best suited. It provides an out-of-the-box solution but with a high level of freedom in installing and developing new software that will run inside the motes. Therefore MoteWorks is the primary choice for the software part.

The hardware options get reduced as MoteWorks is chosen. The reason is that only hardware made by Crossbow runs it, so for that reason the hardware option is limited to the IRIS, Mica2 and MICAz nodes.

Representatives from Crossbow recently announced that the Mica2 series is going to be discontinued in the near future and recommend moving to IRIS motes since these are a more up-to-date solution.

As an alternative, either one of the TinyNode motes presents a good choice due to its operating systems support and good radio coverage with a 868 / 915 MHz actuation range that will help significantly in the forest scenario.

6. Application Interface

6.1. Introduction

This chapter is devoted to the preliminary prototyping of the graphical interface of the application. A lightweight solution based on Web technology is proposed in order to delegate all the logic and processing among the application business logic layer and the SemsorGrid4Env architecture components. This design follows the principles to be accomplished by the project and related to the rapid application development on top of the SemsorGrid4Env architecture.

6.2. *Main areas*

The application interface will be divided into three main areas:

- Main menu: Provides access links to the main sections of the application.
- Content menu: Shows a list of options from the previously selected section that modifies the content area.
- Content: It is the area where the information is displayed.

An additional area located in the right part, will show up only when required. It is named context-aware area and will contain a set of *widgets* loaded dynamically according to the main content or according to the interface element chosen, in example, when choosing a node in the map view, a summary of the node will be provided in one of the widgets.

Another functionality which can be provided by the widgets is the possibility of customization. The user will have some degree of freedom to visualize in certain widgets the desired data a not only some presets provided by the system.

The following image represents how clicking a link of an area modifies the pointed section.

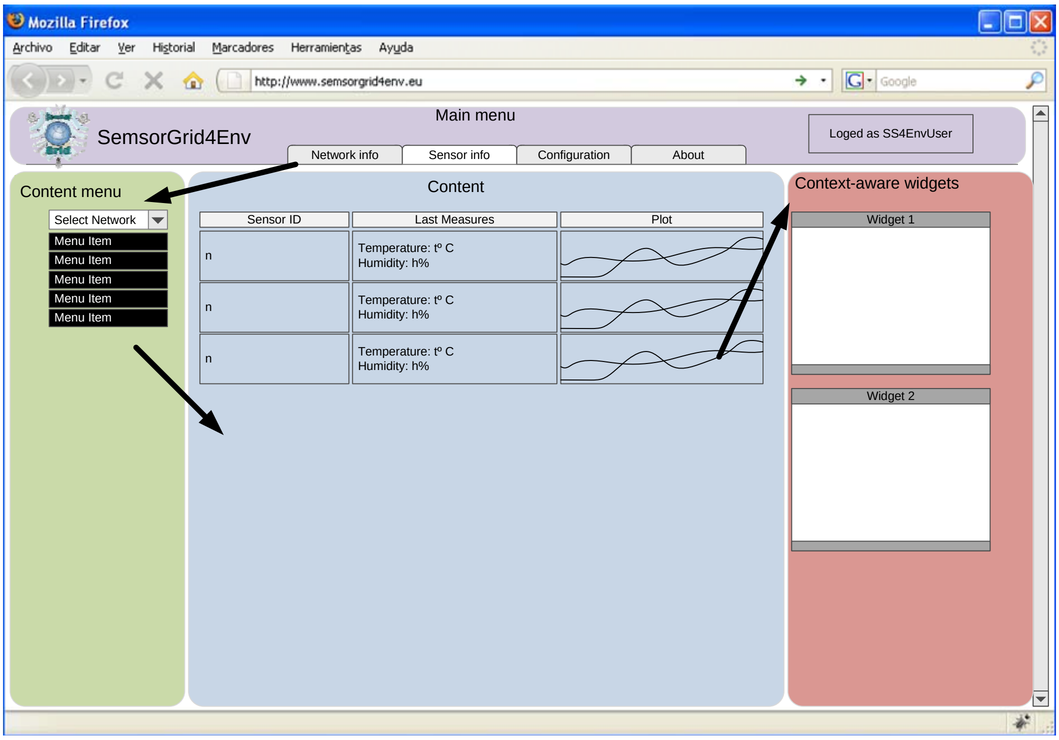

Figure 7: Application, Interface areas

6.3. Network view

Displays the wireless sensor network with different selectable layers depending on the demanding data. Some of these layers will be:

- Network links, showing node's hardware status, such as link quality among nodes, battery levels, etc
- Topographic layer
- Color-coded layers showing the measured variables
- Satellite layer which displays up-to-date information

The last mentioned layer is described more in detail in the section 9 data correlation between the WSN and the satellite data.

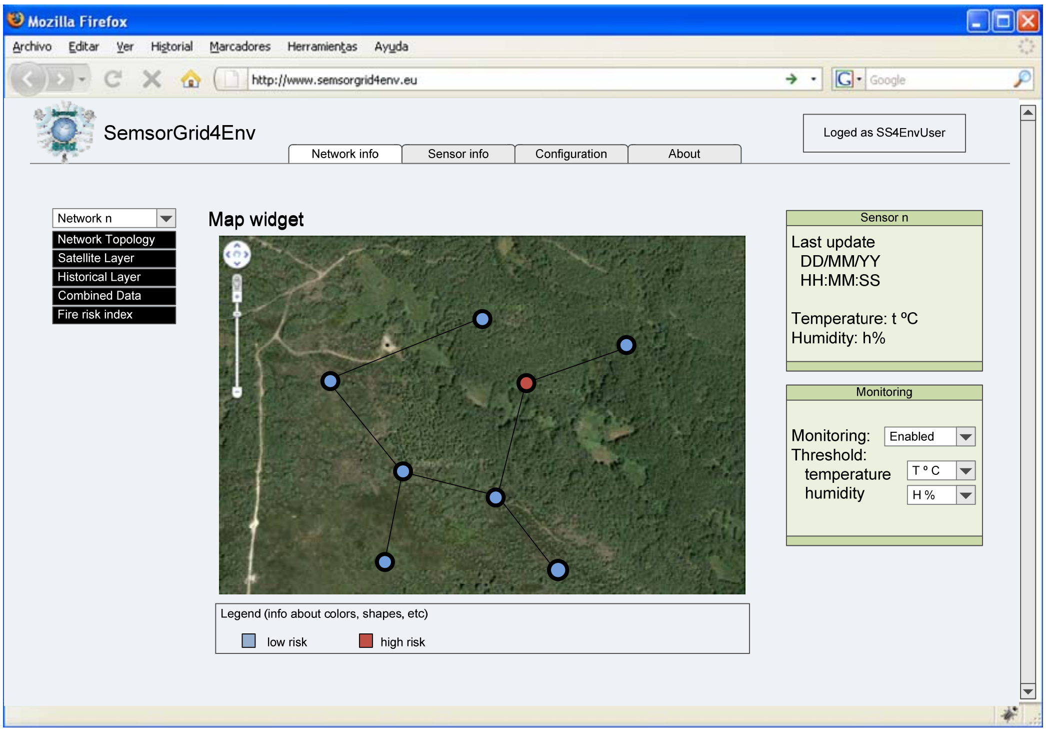

Figure 8: Application, Network view

The following image shows how the WSN-status could be displayed for a quick look instead of using only tabular data.

In the following list and image is shown how different network characteristics could be represented:

- Node size: activity level
- Inner circle: risk level, blue shade represents low risk, red shade shows a potential risk detected by the node
- Outer circle: battery level
- Link weight: link quality
- Link color represented as a shaded gray: link activity

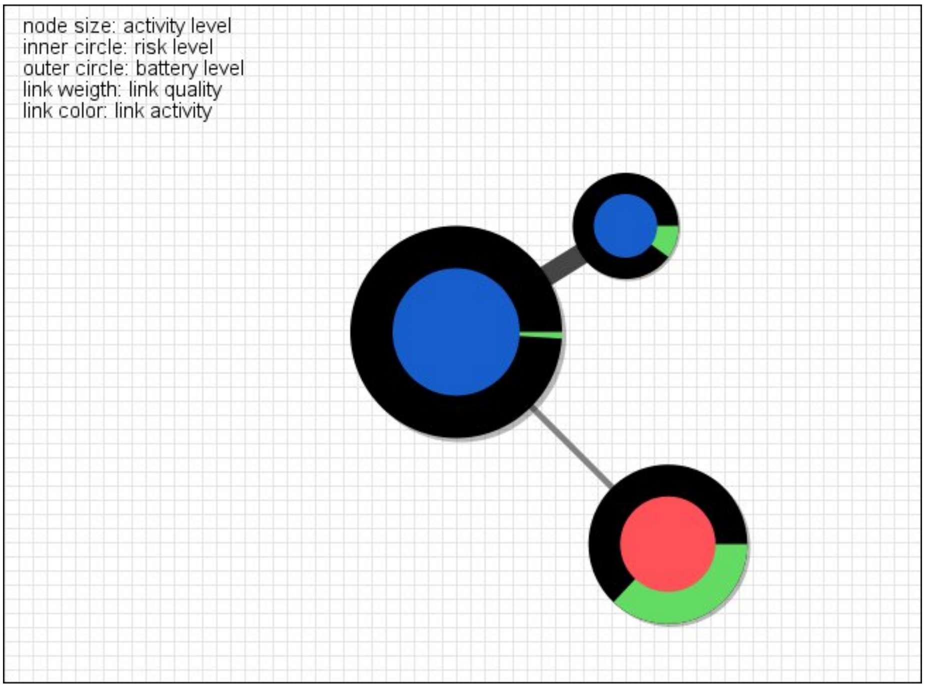

Figure 9: Application, Network detail

6.4. Node view

This section is a shortcut to a quick sensor information list, which provides an overview of all or a set of nodes in a network. In future revisions some operators can be included to list some personalized data.

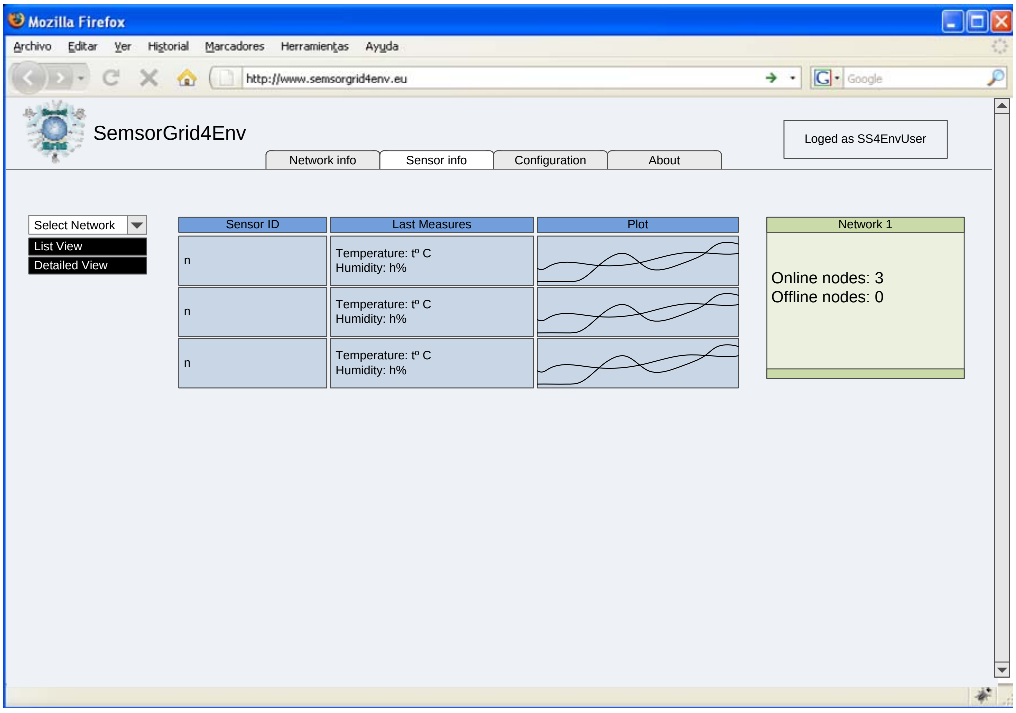

Figure 10: Application, Node view

6.5. Configuration

This section provides a configuration area for users registered in the system. The user will be allowed to configure their own widgets according to their needs. The widgets will show information retrieved for a set of method which will query the system.

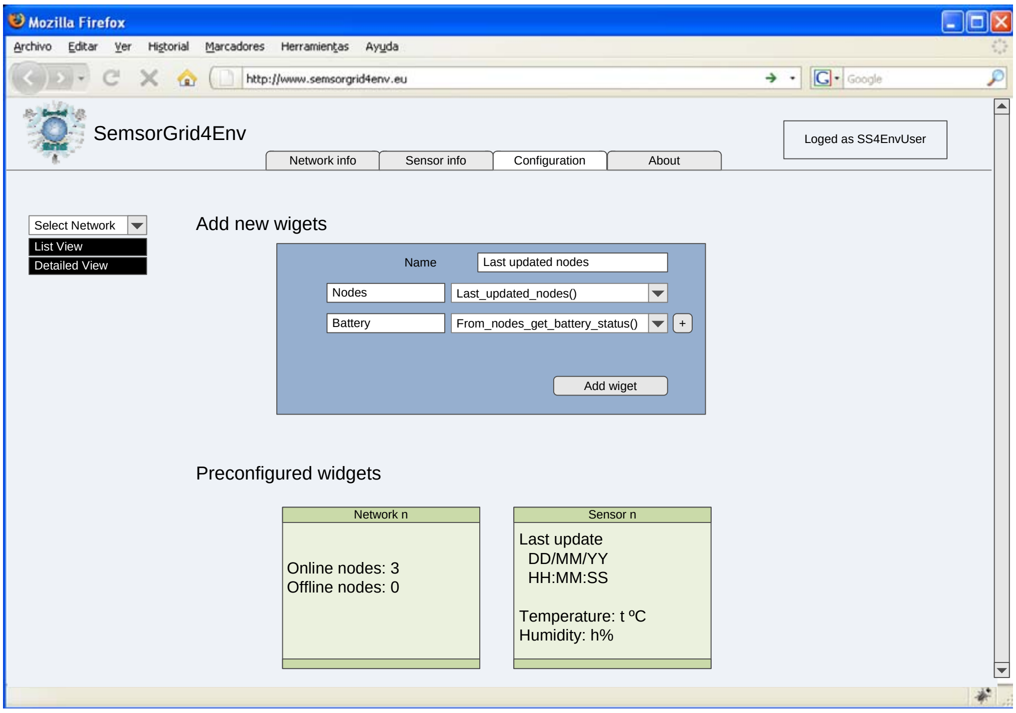

Figure 11: Application, Configuration

6.6. About

This section provides information about the project itself, the process involved, as well as the partners and technologies used and developed.

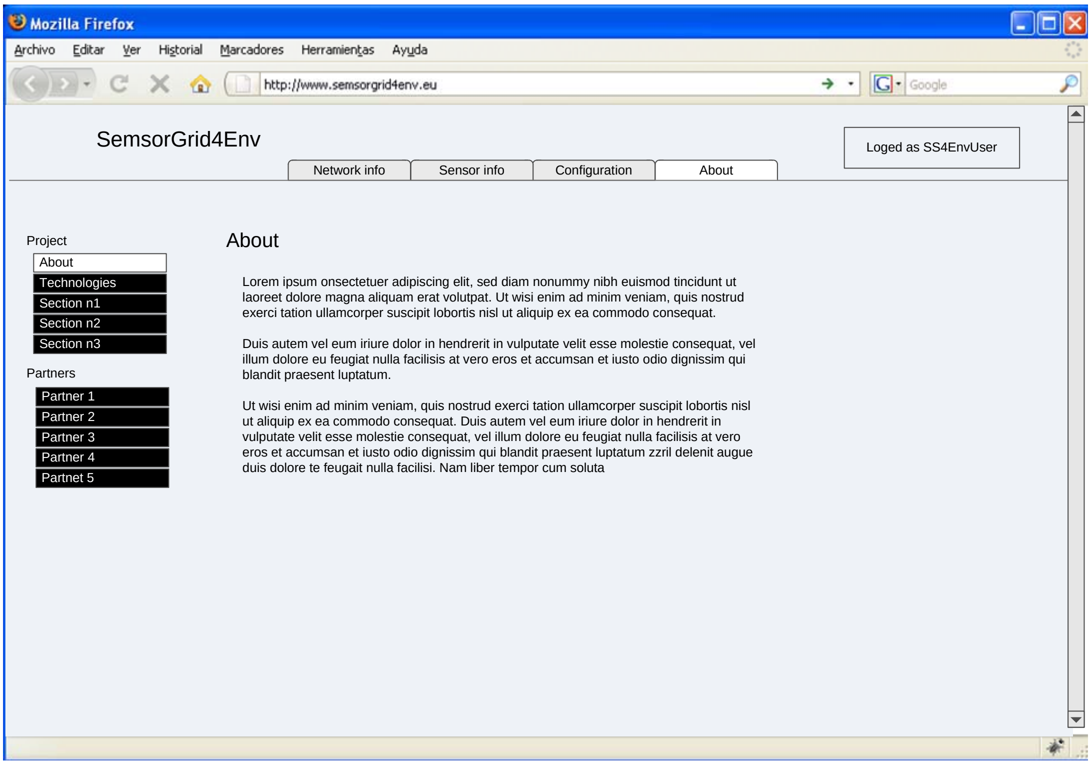

Figure 12: Application, About section

7. Fire application's use cases

7.1. Purpose

This task is related to the specification of application's UML use cases that come up from the previous requirements. The use cases described in this section are to be used only for the identification of architecture interfaces and basic functionalities.

These cases summarize the basic functionality of the application regarding the architecture interfaces and are referred basically to:

- The start-up of the application and the initial deployment of the sensor network.
- The configuration of the application in order to how the application receives useful and user-friendly information from the architecture.
- The configuration of the application in order to facilitate the user the task of monitoring the fire risk and its prevention.
- The retrieval of historical data related to observations measured by the sensor network and historical data from satellite images.

7.2. Specification

The illustration below depicts the main use cases of the application. There is only one actor, the user that operates the application.

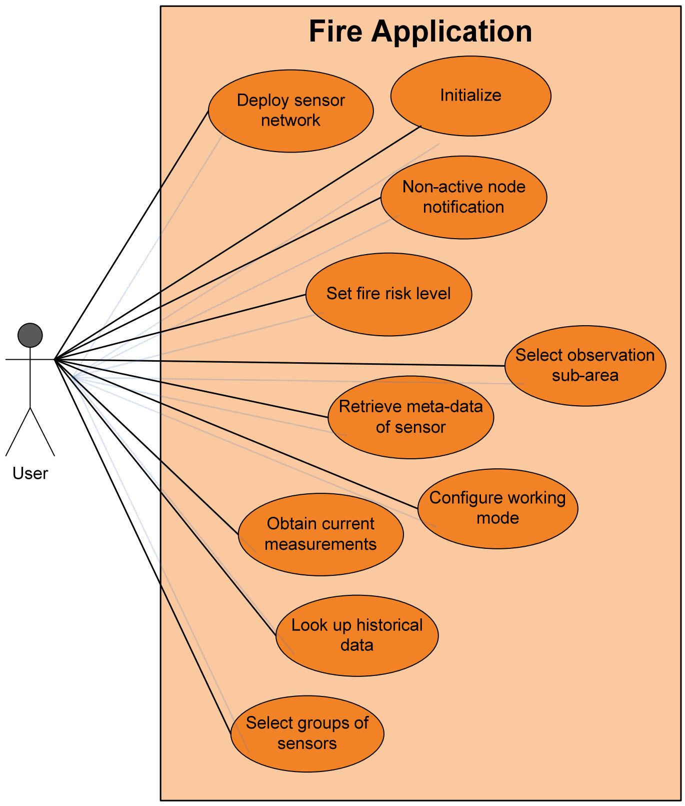

Figure 13: UML use cases

7.3. Assumptions

The following use cases are made taking into account these basic assumptions:

- The application manages two kinds of queries in terms of the kind of information provided by these queries:
    - Operational query: this kind of query collects real-time information observed by the sensor network. There's only one operational query running at the same time collecting and sending information to the application: this is the so-called active query.
    - Historical query: this kind of query collects historical information observed by the sensor network and also collects information contained into the satellite images.
- The application can use different information sources to transform and enrich information coming from data services. At least, the application needs the following functionalities:
    - *Sensor network observations:* The raw observations coming from the network can be represented visually. A range of colors and sort of icons associated with measurement values could make an initial representation of these raw observations.
    - *Sensor grouping:* it's responsible for the management and representation of the raw observations coming from the network grouping this information based on the similarities of measurement values.
    - *Satellite image function:* It is responsible for the management and representation of the data coming from satellite images.
    - *Combined information functionality:* it's responsible for the management and representation of the data coming from the sensor network combined with the data coming from the satellite.
    - *Historical sensor network:* It is responsible for the management and representation of the historical data related to sensor networks. The management of this sort of data will cope with aggregation operators. The historical sensor network data can be retrieved for a defined time period and can be used for trends analysis and extrapolation.
    - *Historical satellite image representation:* It is responsible for the management and representation of the historical data related to the satellite images. The management of this sort of data will cope with aggregation operators.
      The historical satellite data can be retrieved for a defined time period and can be used for trends analysis and extrapolation.
    - *Combined historical information functionality:* It is responsible for the management and representation of the historical data provided by the sensor network combined with the historical data provided by the satellite.

7.4. *Uses cases*

7.4.1. Use case: Initialize

Use case 0010: Initialize
ID: CU-0010
Purpose: The application starts and initializes its environment and the communication with the architecture and the sensor network.
Actors: User
Pre conditions: The network must be deployed.
Events flow: ❑ The user starts the application. ❑ The application finds the sensor network thanks to its metadata that was recorded previously in the registry, accessible through middleware. ❑ The application gets the active query previously saved in the middleware. ❑ The application gets the current observations measured by the network. ❑ The application subscribes to the events generated by the active query. ❑ The application waits for the events to come.
Post conditions: ❑ The application establishes a communication with the middleware with a specific kind of communication (method call or event subscription). ❑ The application shows the default layer. ❑ The application updates its state and renders the data over defined layers taking into account the information thanks to the execution of the operational query inside the network and the combination of these observations with the satellite image.

Table 11. 'Initialize' use case.

7.4.2. Use case: Deploy sensor network

Use case 0020: Deploy sensor network

Use case 0020: Deploy sensor network
ID: CU-0020
Purpose: The user deploys the sensor network, so the architecture stores the metadata associated with the network and provided by the application.
Actors: User
Pre conditions: ❑ The application has the metadata used to register and deploy the network. ❑ The application has a preconfigured working mode.
Events flow: ❑ The user chooses the associated action to deploy the sensor network. ❑ The application builds the metadata associated with the network and sends it to the middleware to be saved in the registry. ❑ The application builds an operational query according to the default constraints through the middleware. ❑ The application sets the operational query as the active query.
Post conditions: ❑ The application waits for information coming from the middleware according to the new deployment. ❑ The application updates the information according to the data taken from the registry.

Table 12. 'Deploy sensor network' use case.

7.4.3. Use case: Working mode configuration

Use case 0030: Working mode configuration
ID: CU-0030
Purpose: The application sets one of the data retrieval methods, manual or automatic fired by an event.

Use case 0030: Working mode configuration
Actors: User
Pre conditions: ❑ The network must be deployed.
Events flow: ❑ The user chooses the desired working mode for fetching data from the WSN. ❑ The application discovers through the middleware the sensor network using its metadata. ❑ The application gets the active query previously saved in the middleware. ❑ The application subscribes to an event if event mode is chosen or directly fetches the node data if manual mode is selected. ❑ The user can unsubscribe to an event.
Post conditions: ❑ The application enters to a waiting mode if event mode is chosen. ❑ The application shows in the information area which configuration mode is in use.

Table 13. 'Working mode configuration' use case.

7.4.4. Use case: Retrieve sensor metadata

Use case 0040: Retrieve sensor metadata
ID: CU-0050
Purpose: The application retrieves information about the sensor node and represents it graphically.
Actors: User
Pre conditions: ❑ The network must be deployed.

Use case 0040: Retrieve sensor metadata
Events flow: ❑ The application discovers through the middleware the sensor network using its metadata. ❑ The application gets the sensor node metadata provided by the WSN.
Post conditions: The application will use the metadata in further operations and as well can update its visual components taking into account the metadata received.

Table 14. 'Retrieve sensor metadata use case.

7.4.5. Use case: Set fire risk level

Use case 0050: Set fire risk level
ID: CU-0050
Purpose: The application can set a threshold value in order to set a fire risk level. When the registered values are higher than the threshold an alarm will be triggered.
Actors: User
Pre conditions: ❑ The network must be deployed.
Events flow: ❑ The application discovers through the middleware the sensor network using its metadata. ❑ The user sets the threshold risk level for the complete WSN or for a group of nodes (threshold of the values of physical measurements).
Post conditions: The application enters in a surveillance mode waiting for incoming warnings.

Table 15. 'Set fire risk level' use case.

7.4.6. Use case: Non-active node notification

Use case 0060: Non-active node notification
ID: CU-0060
Purpose: The application gets notified asynchronously when a node is not active.
Actors: User
Pre conditions: ❑ The network must be deployed.
Events flow: ❑ The application discovers through the middleware the sensor network using its metadata. ❑ The application subscribes to the registry for receiving updates of the network state. ❑ The registry notifies a change in the network and sends the update to the registered application.
Post conditions: The application receives the current measures and show them graphically if it is desired. In case of node failure, the registry will notify the state to the Data Access Service which will be in charge to filter the data propertly.

Table 16. 'Non-active node notification' use case.

7.4.7. Use case: Obtain current measurements

Use case 0070: Obtain current measurements
ID: CU-0070
Purpose: The application retrieves current measurements for given node in the sensor network.
Actors: User

Use case 0070: Obtain current measurements
Pre conditions: ❑ The network must be deployed.
Events flow: ❑ The application discovers through the middleware the sensor network using its metadata. ❑ The application obtain data access ❑ The application gets the current observations measured by the network.
Post conditions: The application receives the current measures and shows them graphically if it is desired.

Table 17. 'Obtain current measurements' use case.

7.4.8. Use case: Select observation sub-area

Use case 0080: Select observation sub-area
ID: CU-0080
Purpose: The user selects the sub-area up to be observed, so the application only receives information concerning this sub-area.
Actors: User
Pre conditions: ❑ The network must be deployed.
Events flow: ❑ The user chooses the associated action to select the observation sub-area. ❑ The application discovers through the middleware the sensor network using its metadata. ❑ The application modifies, through the middleware, the active query that only collects observations coming from sensors into the selected sub-area.

Use case 0080: Select observation sub-area
Post conditions: ❑ The application only receives information coming from sensors situated in the selected sub-area. ❑ The application only updates its visual components associated to the sub-area.

Table 18. 'Select observation sub-area' use case.

7.4.9. Use case: Look up historical data

Use case 0090:Look up historical data
ID: CU-0090
Purpose: The user looks up the historical data stored by the architecture.
Actors: User
Pre conditions: ❑ The network must be deployed.
Events flow: ❑ The user chooses the associated action to retrieve historical data related to measurements coming from the network, from the satellite images or combined historical data. ❑ The application builds through the middleware the query to retrieve the historical data.
Post conditions: ❑ The application receives the historical data derived from the defined query. ❑ The application updates its layers taking into account the historical data received.

Table 19. 'Look-up historical data 'use case.

7.4.10. Use case: Select group of sensors

Use case 0100: Select group of sensors
ID: CU-0100

Use case 0100: Select group of sensors
Purpose: The user groups the sensors, so the architecture sends to the application the observations grouped by similar measurement values.
Actors: User
Pre conditions: ❑ The network must be deployed.
Events flow: ❑ The user chooses the associated action to group sensors. ❑ The application discovers through the middleware the sensor network using its metadata.
Post conditions: ❑ The application updates its layers and visual components considering the new information received.

Table 20. 'Select group of sensors' use case.

8. Fire application's sequence diagrams

8.1. Purpose

This task is related to the specification of the UML sequences diagrams derived from the previous UML use cases.

8.2. Specification

The specification is subject to the following assumptions.

8.2.1. Assumptions

The following sequence diagrams are made taking into account these basic assumptions:

The architecture has the following components:

- Middleware
- Registry
- DataAccess

The responsibilities of each component are:

- Middleware: it manages the communication between end applications and the rest of the components of the architecture. It stores queries defined by the end applications and the subscriptions to events made by the applications.
- Registry: it stores the metadata associated with each sensor network.
- DataAccess: it manages the data coming from heterogeneous sources like:
  - Sensor Networks.
  - Satellite image repository.
  - Historical data stored in databases.

Assumptions for the application:

- The application will be able to define and store different sort of queries in the middleware.
- The application will be able to set an active query. This query will be responsible for providing information to the application until another query will be set.
- The application will be able to subscribe to events derived from the execution of the defined queries.

8.2.2. Sequence diagram: Initialize

This sequence diagram corresponds to the *Initialize* use case.

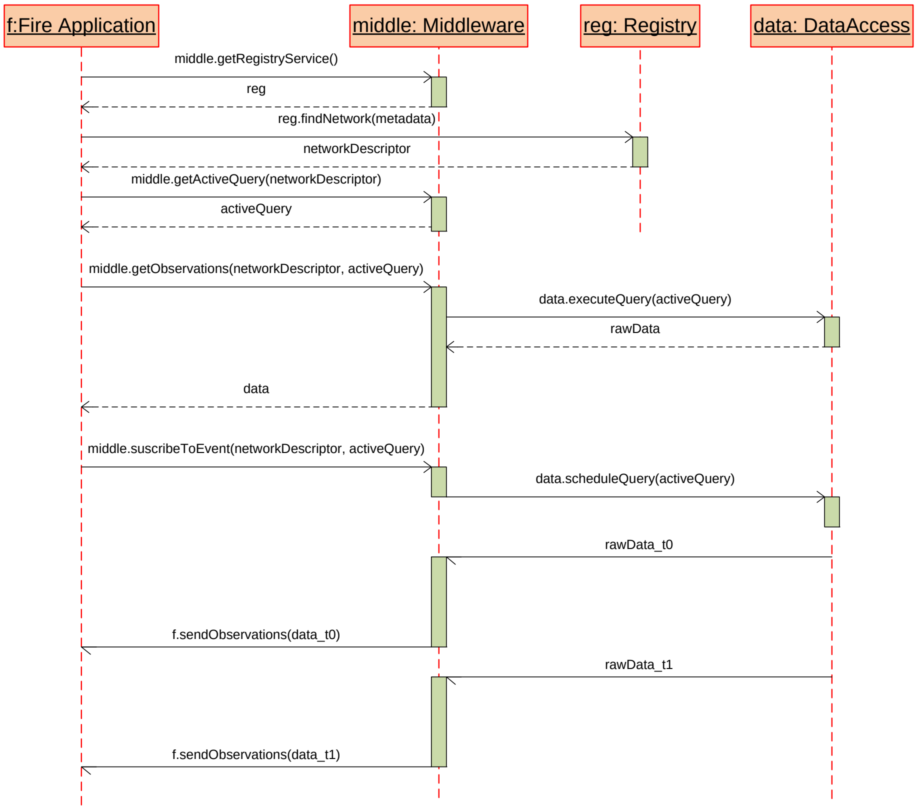

Figure 14: 'Initialize' sequence diagram

8.2.3. Sequence diagram: Deploy sensor network

This sequence diagram corresponds to the *Deploy sensor network* use case.

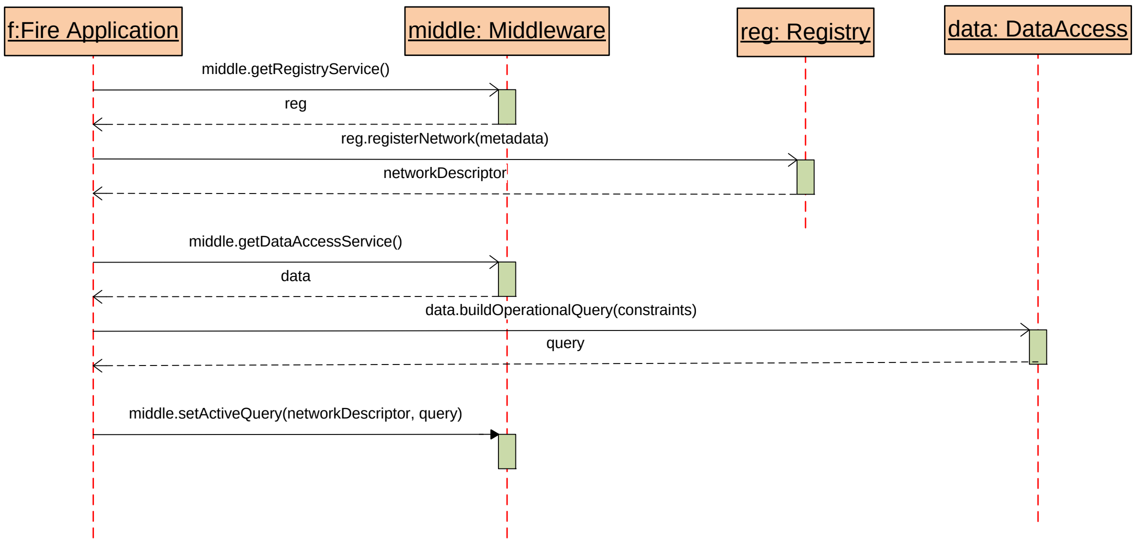

Figure 15: ‘Deploy sensor network’ sequence diagram.

8.2.4. Sequence diagram: Configure working mode

These sequence diagrams correspond to the *Configure working mode* use case in manual and surveillance mode respectively.

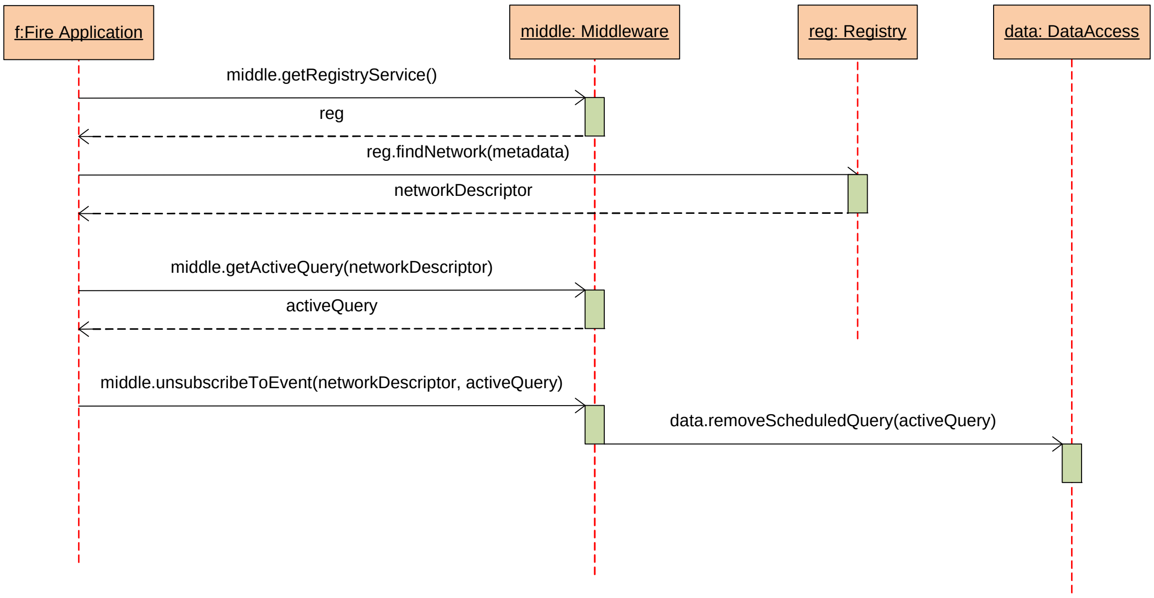

Figure 16: ‘Configure working mode’ (manual mode) sequence diagram

.

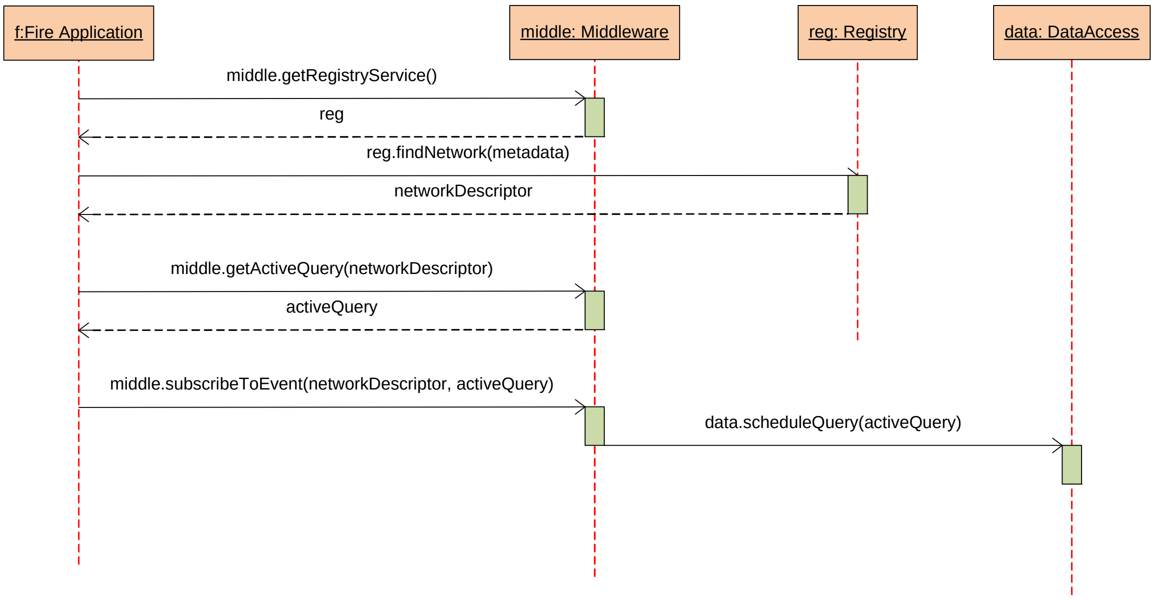

Figure 17: ‘Configure working mode’ (surveillance mode) sequence diagram.

8.2.5. Sequence diagram: Retrieve sensor metadata

This sequence diagram corresponds to the *Retrieve sensor metadata* use case.

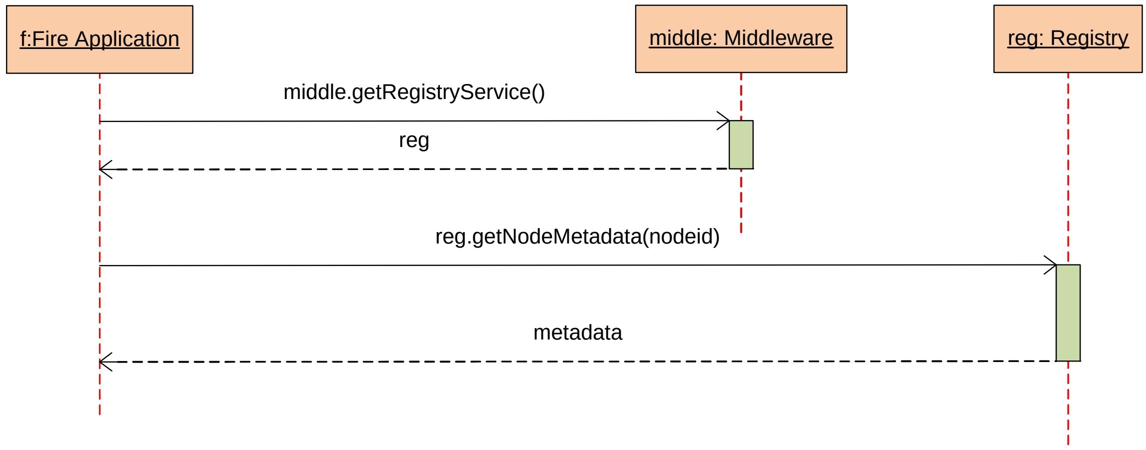

Figure 18: 'Retrieve sensor metadata' sequence diagram

8.2.6. Sequence diagram: Set fire risk level

This sequence diagram corresponds to the *Set fire risk level* use case.

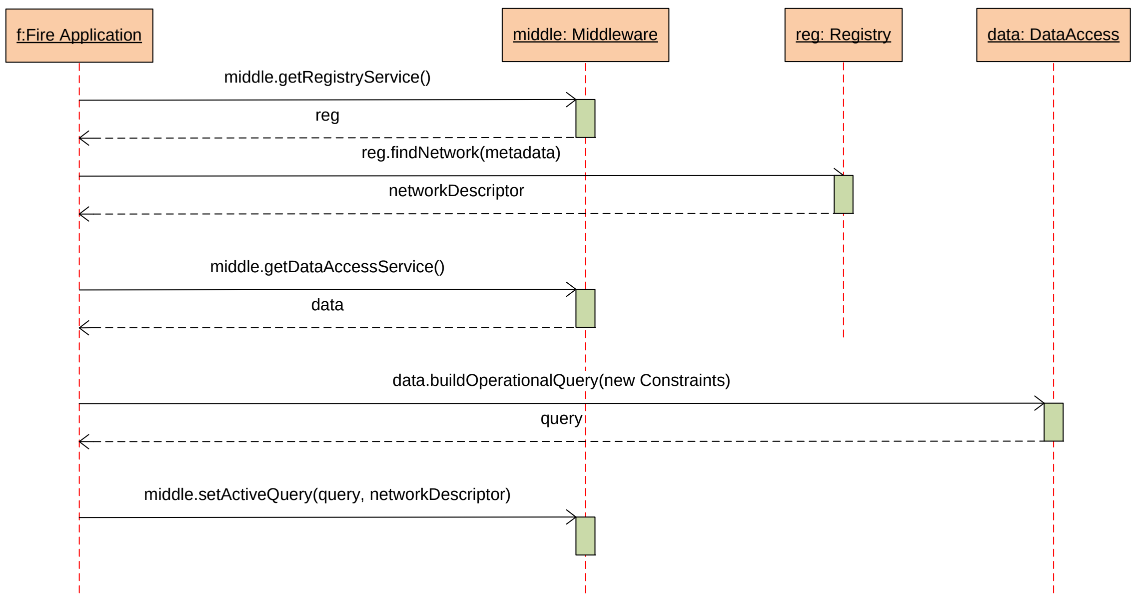

Figure 19: ‘Set fire risk level’ sequence diagram.

8.2.7. Sequence diagram: Non-active node notification

This sequence diagram corresponds to the *Non-active node notification* use case.

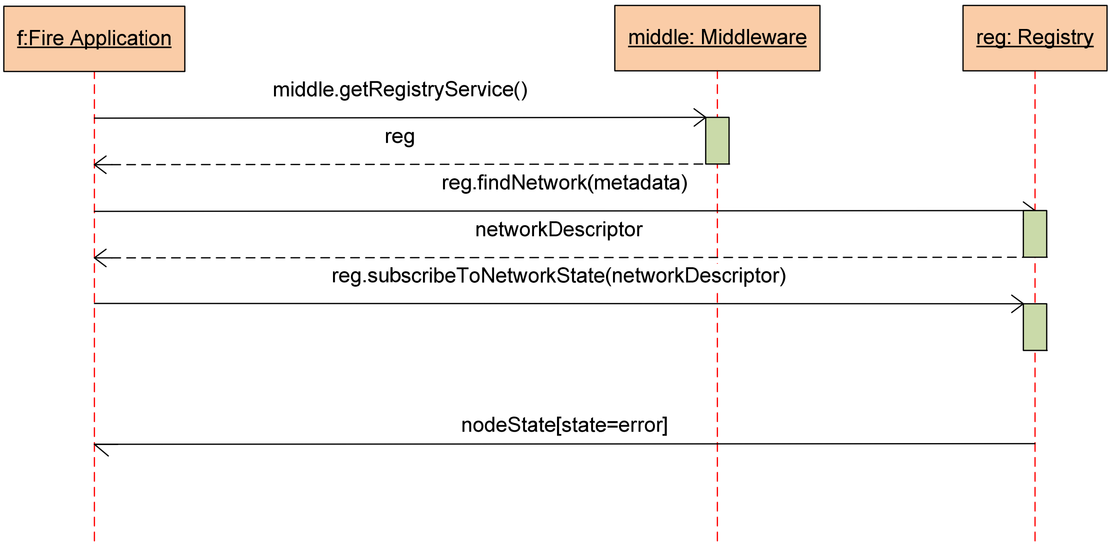

Figure 20: ‘Non-active node notification sequence diagram.

8.2.8. Sequence diagram: Obtain current measurements

This sequence diagram corresponds to the *Obtain current measurements* use case.

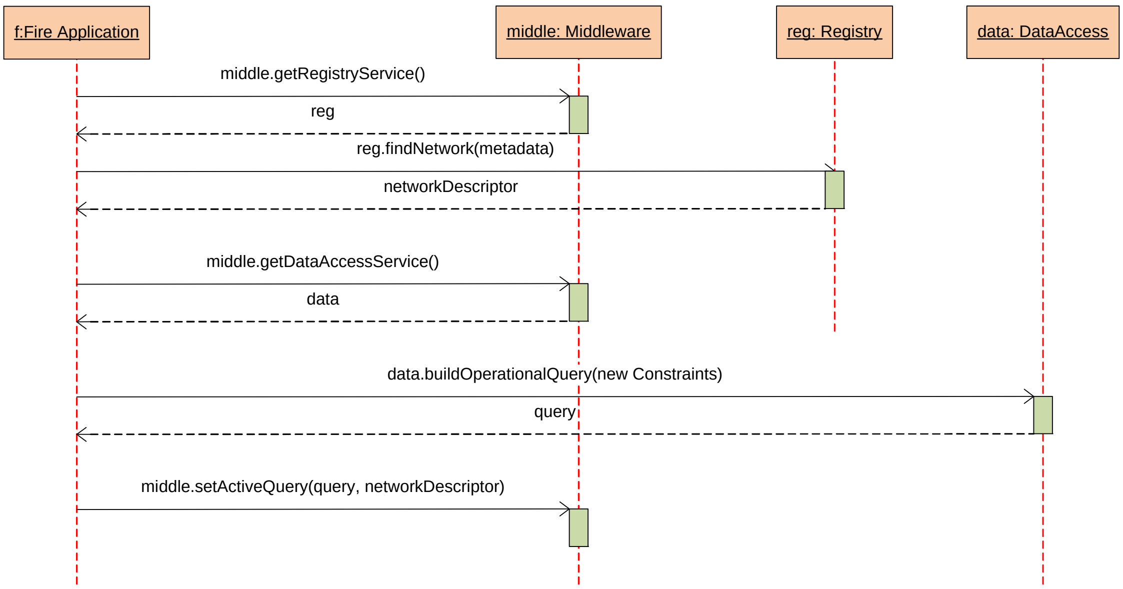

Figure 21: ‘Obtain current measurements’ sequence diagram.

8.2.9. Sequence diagram: Select observation sub-area

This sequence diagram corresponds to the *Select observation sub-area* use case.

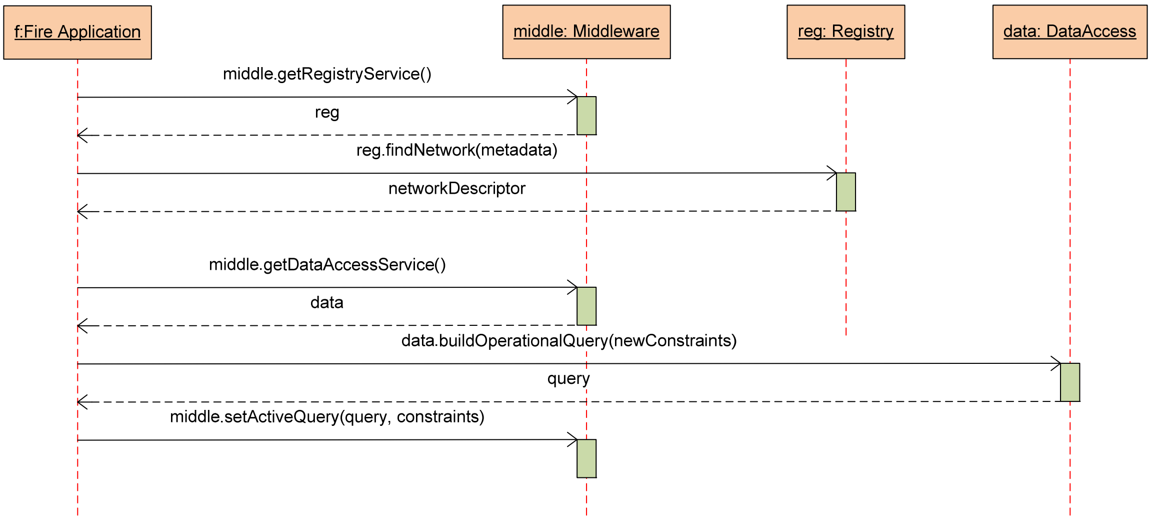

Figure 22: 'Select observation sub-area' sequence diagram.

8.2.10. Sequence diagram: Look up historical data

This sequence diagram corresponds to the *Look up historical data* use case.

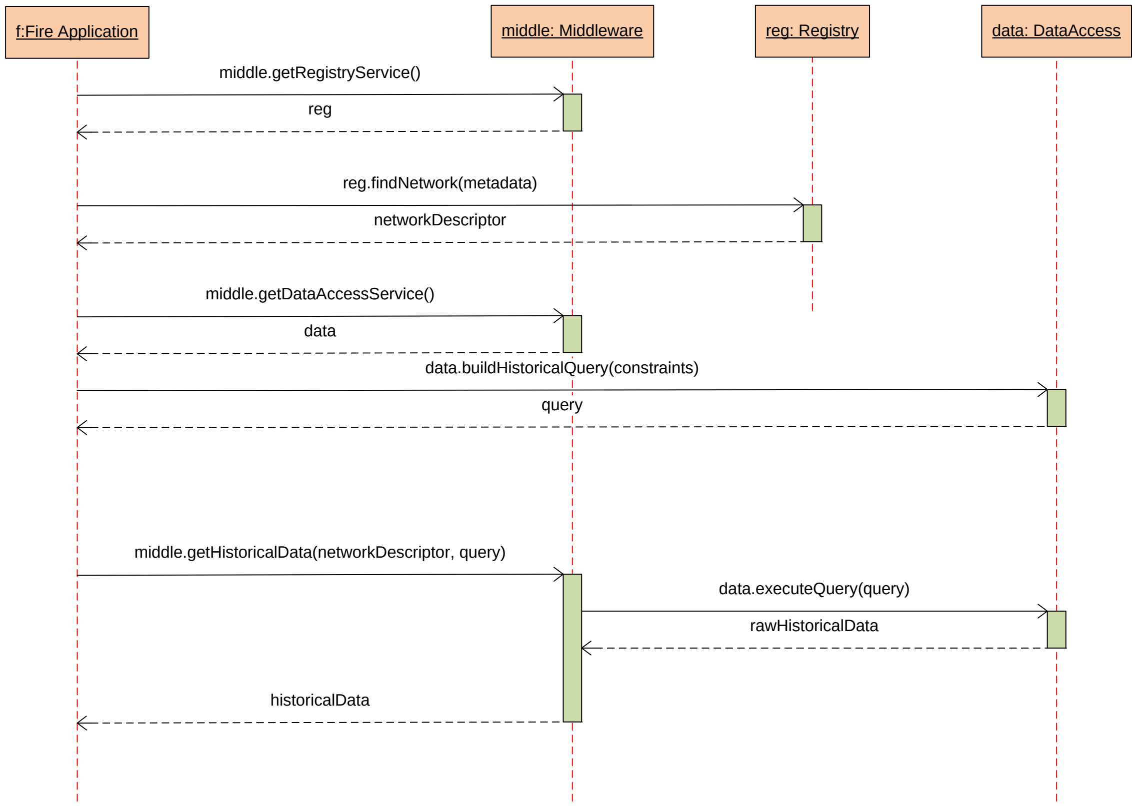

Figure 23: ‘Lookup historical data sequence diagram.

8.2.11. Sequence diagram: Select group of sensors

This sequence diagram corresponds to the *Group sensors* use case.

Figure 24: ‘Select group of sensors’ sequence diagram.

9. Conclusions

This report intends to identify and specify the requirements for the fire use case application responsible for forest fire prevention. These requirements come up from the needs of the stakeholders and final users in order to ease the monitoring through the application, from the characteristics of the physical sensors and nodes to be used, from the real scenario in which the sensor network will be deployed and the basic functionalities the architecture must provide.

The requirements regarding the physical sensors and the nodes of the sensor network are identified from a previous market research and the constraints related to the in-network process approach that will be considered in the fire use case. A remarkable constraint is the necessity of the node to have a certain software platform (like tinyOS or Contiki) for running the algorithms responsible for taking the desired data from the network.

It's important to highlight that this report only includes the application's requirements concerning the operations the user will have for the monitoring task. It doesn't focus on requirements related to the final look & feel of the application, neither on the requirements of the algorithms to combine the data from the sensor networks and the data of the satellite images.

From these requirements the main UML use cases of the application that demand functionalities from the architecture have been derived and included in this report. The specification of these use cases is justified for clarifying and establishes a basic interface between the application and the architecture through the middleware, which is the component that interacts with the application and the rest of the components of the architecture like the registry and the data access component.

Following the use cases of the application are the UML sequence diagrams that describe the interaction between the application and the components of the architecture for each UML use case. The specification of these diagrams is subject to the hypothesis regarding what components the architecture will have and the responsibilities and functionalities each component will offer.