A Platform for Wildfire Fighting: A Comprehensive Approach

David Villa, Félix J. Villanueva, Maria J. Santofimia, F. Moya, and Juan Carlos López

Department of Information Technologies and Systems
School of Computer Science
Ciudad Real, Spain
david.villa@uclm.es

Abstract. This paper describes a support platform for integrating services and technologies, most of them specifically devoted to wildfire fighting. The platform presented summarizes a three years work project, funded by the Prometeo research project, involving a consortium of fifteen companies and several research centers and universities. Sensors, control command centers, 3D visualization technologies, fire real-time simulations, fire fighting helicopters and airplanes, or fire fighters, among the most relevant, are examples of the entities integrated into a comprehensive approach whose main purpose consists in avoiding the "information island" phenomenon, currently dramatically impacting the wildfire fighting task.

1 Introduction

Standard procedures to wildfire fighting are characterized by lack of precision and delays of verbal communications held by the extinction coordinator and the different teams involved in the field works. Despite the fact that there might exist information gathering subsystems such as weather historic data, weather forecast or satellite images, this information can be considered obsolete taking into account the period of measurement in contrast to how quickly wildfires evolve.

This type of scenarios, in which there is an evolving event that somehow can be predicted, could be benefit from a platform that provides real-time live data, based on sensed data, simulations and forecasts. However this type of architecture poses highly demanded challenges such as integration issues among heterogeneous systems and devices as well as communication support for the different systems sensors and devices involved in this type of scenarios.

Information representation is also a key issue, since data is coming from different sources, implementing different standards or information models. Centralizing this information into sink is also a relevant need that should be tackled.

Wildfire fighting is a key concern for the Spanish government. Every summer a sequence of catastrophic fires ruin large extensions of forest, having a dramatic impact in the local fauna, sometimes even getting to populated areas. Due to the economic and environmental impact that fires have on countries, the Spanish government has funded a four years research project intended to leverage technological advances to support fire extinction works.

This project has been mainly devoted to provide fire extinction groups with updated and live data about how fire and weather conditions evolve. Additionally, this information can be combined with climate and fuel models to simulate the directions towards which the fire is evolving. This main intention is articulated through several subgoals, which basically consist in:

- A permanent infrastructure to manage incoming wildfire alarms and sighting.
- Integration and coordination of a wide variety of live information providers.
- Full featured graphical representation for all of the information collected.
- Mechanisms to create ad-hoc communication facilities on the event of a wildfire.
- Data representation homogenization to make parties inter-operate.
- Common event format specification for all parties.

The rest of the paper is organized as follows. Section 2 presents similar works focused in communication infrastructure and details, section 3 describes the project overall goals and components; then section 4 deals with the communication requirements and the proposed approach.

By means of a communication middleware, platform provides a powerful integration mechanism which is especially important in this scenario due the diversity of programming languages, computer architectures and even partners staff technical knowledge background.

2 Related work

The literature revision on firefighting brings into light a relevant number of works, most of them mainly focused in a specific domain. Wildfire monitoring [6] [7], architectures for firefighting tactical training [3], task allocation for brigades [4], autonomous fire-fighting robot design [5], etc. are examples of works in which the emphasis is made in a specific domain rather than considering the overall complex problem of firefighting. Despite the fact that this approach enables a better understanding of the different problems faced by different domains, the lack of comprehensive approaches leaves numerous issues open that would otherwise arise during the integration of partial solutions.

Several works basically focus their proposal in using new IT technologies for gathering information for wildfire prediction, detection and/or monitoring. An example of this type of works can be found in [2] in which a 802.15.4 based wireless sensor network (WSN) is designed, basically for indoor environments. The platform also includes RFID tag position (sensors, firefighting men, etc.) and video surveillance cameras.

Additionally, other works base their proposals in integrating high level information. In [1], a Web-Service based architecture is proposed for integrating information systems in an abstract, loosely coupled and coarse-grained way. The main drawback of this work is that it focuses in high-level information services, involving the use of Web-Service protocols (e.g. XML-text based protocols, HTTP, etc.). However, these type of protocols are not appropriate for low-bandwidth networks used in sensor networks, delay tolerant networks, etc., which will be common in future fire fighting scenarios as we will see later.

An additional approach encompasses those efforts aimed at integrating specific tools. For example in [8], an integration of simulations with geographical information systems (GIS) is described, enabling data representation through visualization improvements. Moreover, the work in [9] describes an approach for integrating environmental models like satellite images, meteorological information, etc. None of these works consider the integration of environmental real-time information, brigades position, etc. essential for a successful approach to wildfire firefighting.

The literature revision therefore brings into light a lack of comprehensive approaches that support the integration of information, coming from different sources and devoted to assist in tactical decisions in a wildfire extinction. From an engineering point of view and as we will see later, we followed a bottom-up approach identifying data sources relevant to the purpose of the system and integrating all elements in a seamless way. When we started the project, we analyzed firefighting technologies used in fire extinction, identified future technologies to be integrated and started to extract the information model from acknowledge experts.

3 Project overview

One of the main contributions of the Prometeo project is the multidisciplinary approach it advocates for managing the heterogeneous sources of information and actuation. In this sense, different people roles and devices are involved in the task of compiling wildfire live information. This information is provided to the extinction coordinator who will be able to take grounded decisions, that will be strategically aimed at protecting people, goods and infrastructure, and natural resources, in this order. Providing the extinction coordinator with live and simulated information about the current and future state of the fire does not only minimizes response times but it also helps on preventing irreparable damages. For these reasons, Prometeo may be seen as a real-time Decision Support System.

In order to tackle this general goal, the different stages that are traditionally involved in a wildfire extinction have to be adopted and adapted to incorporate the technological and communication advances considered in this project. A summarized version of the involved stages is described underneath. Additionally, figure 1 depicts the different elements involved in the following stages:

- The PSU (Prometeo Support Unit) receives a fire alarm that specifies the approximate location where the fire has been detected. The fire alarm might come from different sources: deployed sensors, observation towers, satellite images, etc.

– A helicopter takes off as soon as the fire alarm has been validated. It flies to the given location waiting for further information to be received during fly time. This helicopter carry three people: the person who operates the camera and two transmission operators, apart from the people operating the helicopter.
– In the meantime, the *support unit* gathers preliminary weather forecast. This information is combined with latest news and events that somehow helps on completing a more accurate view of the overall wildfire situation.
– The *support unit* is constantly updating the helicopter crew with new information, using a 3G connection.
– The helicopter takes ground in a safe area near the wildfire. There, a transmission operator lands, carrying the required equipment to establish a PAU (Prometeo Advanced Unit), which can be seen as the communication center of the *ad-hoc* infrastructure that it is being deployed. The *advanced unit* includes an antena to establish a satellite link to connect to the *support unit*.
– The helicopter takes off and explores the surrounding area taking photos, infra-red (IR) and multi-spectral images. That information is sent back to the PUA using WIMAX/VHF connection. The *advanced unit* forwards relevant information to the *support unit* through the satellite link.
– The helicopter might land again to deploy additional sensors and repeaters in those places that become relevant for the fire evolution in which no previous sensors were deployed. These new sensors and repeaters will be in charge of sending data back to the *advanced unit*. The second transmission operator left back to the *advanced unit*.
– Land brigades arrive to the wildfire scenario using ground vehicles. During their actuation in the field, land brigades are monitored by collecting physiological data using wearable sensors. Environmental data is also collected and combined with the physiological data to be sent back to the *advanced unit*. This information will be used to preserve their safety anticipating risk.

After having deployed the human, technological, and communication infrastructure described above, the fire extinction works start up. Different aspects need to be considered during this process, such as weather, human brigades and helicopter locations, foreseeable evolution of the wildfire, etc. These aspects are monitored and supported on different modules, distributed all along the considered scenario. These modules are described in the following subsections. Additionally, figure 1 represents how these modules are deployed in different locations.

3.1 Environment sensing

Depending on available resources, some “sentinel” sensor nodes (forming a WSN) may be permanently deployed in the forest. These sensors include temperature, humidity, wind speed and direction, intensity of light and rain. These wireless nodes centralize data acquisition on a local bridge that can contact the *support unit* when required through a 3G/GPRS modem attached to the bridge. Data

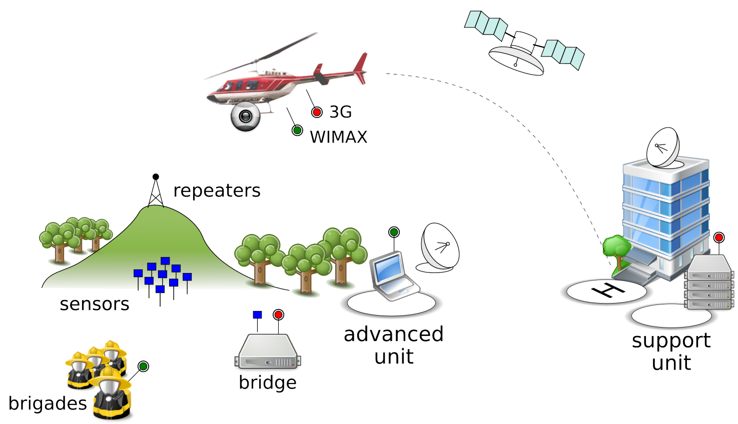

Fig. 1. Prometeo deployment scenario

taken from sensors is locally evaluated to determine risk situations and may lead to a fire alarm event.

At the time of a wildfire, helicopter people may deploy additional sensor nodes and repeaters where required. At that situation, data communication among the WSN and *support unit* can be done through the satellite link, to avoid coverage and bandwidth issues.

Additionally, during a wildfire situation it is desirable to gather information as representative as possible about the environmental conditions. In order to do that, new sensor nodes and repeaters are deployed *ad-hoc* by the transmission operator, landed from the helicopter.

3.2 Brigade monitoring

This module monitors the physiological evolution of the land brigades such as heart rate, body temperature and sweating. These measures are collected through a set of wearable sensors located in the fire fighter suits. The wearable sensor information along with GNSS coordinates are sent to the brigade leader. This transmission is performed every few seconds using the ANT+ protocol[11] and ZigBee[12]. In this sense, the brigade leader works as a router through which information is sent to the helicopter, whenever it is within the WIMAX range. This information is then forwarded by the helicopter to the *advanced unit*. The role of the helicopter in the transmission process is basically that of a repeater.

Additionally, the leader has some other environmental sensors, such as: temperature, humidity and CO_2. As mentioned before, this data is combined with the data received from his/her peers and re-transmitted.

3.3 Forecasting

Regional governments may provide general forecasting information that may be useful for project activities. Also, a inner process provides specific forecasting for the fire affected area and surroundings with better resolution.

3.4 Propagation maps

By means of simulation it is possible to obtain a fire propagation model. Many factors influence such model; one of most important is the vegetation characterization, specially its moisture (FMC). This process produces maps in NetCDF format that may be directly consumed by other processes.

3.5 Image, IR, multi spectral

Helicopters are equipped with conventional and IR cameras. When the helicopter recognizes the fire scenario, it takes images that are immediately sent to the *advanced unit*, and hence to all interested peers.

3.6 Fire simulation

The fire simulation module takes forecast and FMC data and provides estimations every two hours. Simulations include hourly geographic details about the wildfire advance for the next twelve hours. This data is very important for the coordinator. She decides where to send brigades according to the estimated fire behavior specially when people, housing or infrastructures may be affected.

3.7 3D representation

All the gathered sensing data, forecast, images and simulations are centralized, represented and continuously updated in a 3D graphics application. This application is running in the PCA (Prometeo Advance Command) and also in the *support unit*. The PCA is operated by third party people (i.e. government staff).

4 Integration Platform

Regarding the communication platform requirements, Prometeo is composed of two parts. The first one is the *support unit* including all modules attached to it. The second consists mainly on the *advanced unit* and is deployed on-site wherever the fire occurs. Both must be connected to share information in both directions as soon as they are available. The *advanced unit* communication infrastructure needs to be operational in a few minutes and it must be ready to integrate all other modules. Not all modules are present in all deployments and neither at same time. Infrastructure needs to deal with modules that may appear and disappear several times during the wildfire extinction session.

The communication model adopted in the Prometeo scenario is mainly event oriented. All modules are seen as event channel publishers and subscribers. In fact several of then play both roles. The publish/subscribe model has some interesting features for this environment:

– Information is sent as soon as it is available.
– Only a message is required. Consumers do not need to query producers for new data. It just requires one-way communication.
– Publishers are simpler than conventional servers because they do not need to store data waiting for client queries.
– Publishers and consumers do not known each other. All peers are decoupled, the only remote reference they all share is the event channel.
– The above implies it may be an arbitrary numbers of publishers and subscriber and they may be attached at any moment.

4.1 Event channels

Prometeo needs to deal with data of different nature, mapped to *event types*. However, there are some details common to all of them:

– All events are time-stamped with a POSIX time (1 second precision).
– All events have a time expiration value measured in seconds. After that time, subscribers should consider the event as obsolete information.
– Node-based events are geo-positioned. Usually the 3D position of the sensor is used.
– Node-based events have a unique hierarchical node identifier.
– All scalar measures have a *quality* attribute that can be used to indicate precision or sensor quality.

Identified event types are obviously related to the functional modules. Each event type is uniquely related to a unique event channel. They are the following:

Environment
Includes data from the sensor network. Event message types are: `humidity`, `temperature`, `windSpeed`, `windDirecction`, `luminosity` and `rain`. As we stated above, these events include sensor node position and node identifier, measure timestamp and expiration. For greater orthogonality, the value is stored as a 16 bit integer in all of them.

Brigade
Uses only a kind of message but it is complex due to the fact that it carries information about an entire brigade. It includes unique values for timestamp, expiration, node ID, environment temperature, humidity and CO_2, and also per-person values for body temperature, humidity (sweating) and HRM.

Forecast
There are two types of forecast events: weather data (called *meteo*) and FMC calculations. Because this is a very complex information we decided to send a whole `NetCDF` file as the event payload. `NetCDF` is a widely use file

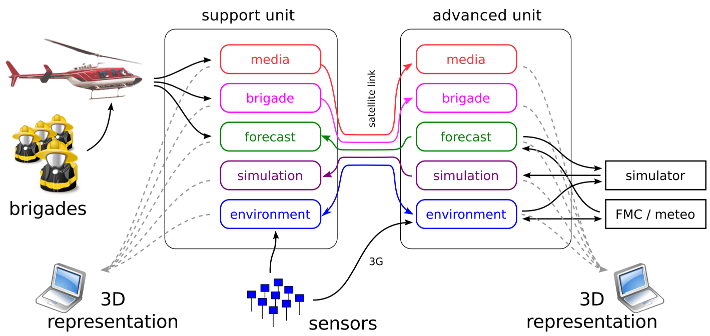

Fig. 2. Event channels

format for that purpose. The file size (around 5 MiB) could be an issue but these events are rare (about one every two hours). Both event types include timestamp and expiration parameters.

Simulation
Simulation results are sent as KML files with size in the range 15-19 MiB. These events are produced at a rate of one every few hours. As forecast events, they have timestamp and expiration.

Media
Cameras installed at the helicopter take image and video sequences and send them to the *advanced unit*. An image may be sent in four different resolutions and three different types: visible image, IR or video fragments. The image is sent in `jpeg` format. Other parameters are helicopter position, timestamp, expiration, camera identifier, image type, resolution and camera orientation.

Almost all modules publish to and consume from at least one event channel. A key issue is that modules are distributed in two different places: *support unit* and *advance unit*, although some modules are present in both. Due to the fact that the connection among these places is realized with a satellite link, it is important to reduce the amount of traffic required. The solution was to install two event brokers (one in each place) and link the corresponding event channels together. This way, a single flow per channel cross the satellite link minimizing the required bandwidth. Subscribers register themselves in the local broker. Figure 2 illustrates the main event channels, their subscribers, publishers and links.

4.2 Event persistence

Logging is a very valuable service in an emergency situation. The middleware infrastructure provides an event logging service that stores all emitted messages in all channels. Despite all events are time stamped, in order to minimize the asynchrony, the logging subscribers add an incoming time tag to each event.

That information is used later to analyze, detect procedure failures, improve fire extinction activities or other forensic purposes. Furthermore, a set of simulated publishers may re-send logged data to reconstruct the whole scenario in real time. This is interesting to test failing modules or to exercise modules not involved in the moment when the fire extinction activities actually occurred.

4.3 Event delay management

The physical medium employed in these type of scenarios, mainly WiMAX, 3G, and satellite, the connectivity and availability issues are not reliable enough to be left unattended. Typical transport protocols do not properly address these issues when connectivity is lost for long enough period of time, in terms of minutes. Due to this fact, it cannot be assumed that a lost of connectivity results in the transport layer discarding messages. In order to overcome this transport layer limitation, the work presented here proposes a solution consisted in temporally store the messages while there is not connection available to resend the awaited messages.

The physical medium employed in this type of scenarios, mainly WiMAX, 3G, and satellite, is not reliable enough to be left unattended. Typical transport protocols do not properly address connectivity and availability issues. When connectivity is lost for a long enough period of time (in the range of minutes) the transport layer would react assuming the connection is lost and discarding messages. In order to overcome this transport layer limitation, the work presented here proposes a solution consisting in temporally storing messages while there is not connection available to resend the awaited messages.

The proposed mechanism is inspired in the Delay Tolerant Network (DTN) paradigm, although shortening the period of time during which messages are kept. In the considered scenario, connectivity lost is not expected to take longer than a few minutes. Additionally, the Prometeo mechanism also differs from DTN in that it provides location transparency. It should be noted that Prometeo messages are addressed to distributed-objects, such as the *event broker*, rather than to host addresses. This feature permits that if the target distributed object changes its location, the message can still properly reach the object. The message identifies the recipient by a unique distributed object identification which is network independent.

5 Prototype

The current Prometeo communication infrastructure is supported by the ZeroC Ice [10]. Ice is a full featured object oriented distributed middleware based on similar principles than CORBA, but substantially lighter and straightforward. Specifically, the IceStorm service is heavily used. IceStorm is a broker based event distribution mechanism. To interconnect channels with same name in the two different places (*support unit* and *advanced unit*) the provided *link* mechanism is applied. To avoid cycles, IceStorm links allow only one hop which is very valuable

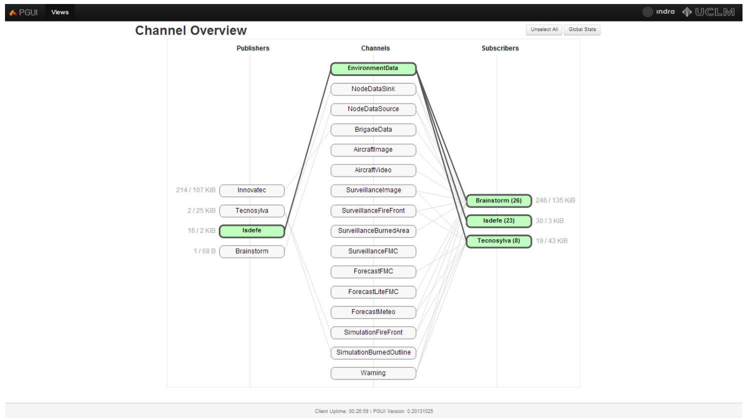

Fig. 3. Event channel monitorization

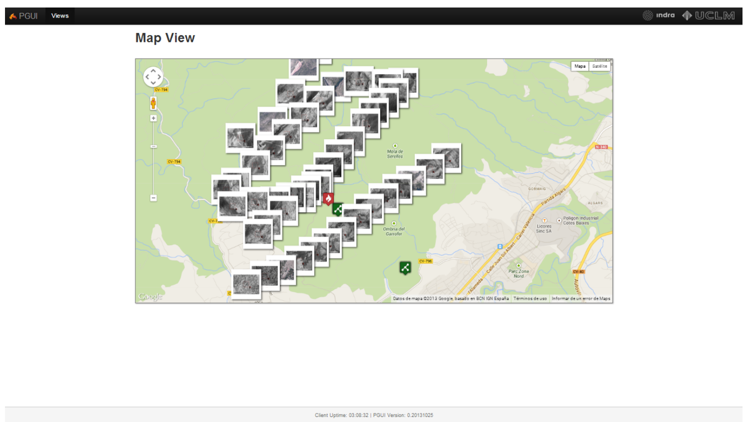

Fig. 4. Geolocalization of pictures taken by the helicopter

in this case. These links are represented in the figure 2 as arrows between the *advanced unit* and the *support unit*.

The modules described here were made available to the project partners. They are implemented using several languages and they run on several operating systems (at least Microsoft Windows and different GNU/Linux flavors). In figure 3 we can see the channel bandwidth used by the partners of the project during final demo in the fire management console.

Some media events are stamped with GPS coordinates so we can see them on the point were they were taken. Figure 4 shows the pictures taken by an helicopter in a path devoted to evaluate the status of the fire.

One of the main objectives of the platform is to feed a fire simulation for predicting the most probable evolution of the fire attending to different parameters. Figure 5 shows a screenshot of one of this fire simulation, the central orange ring

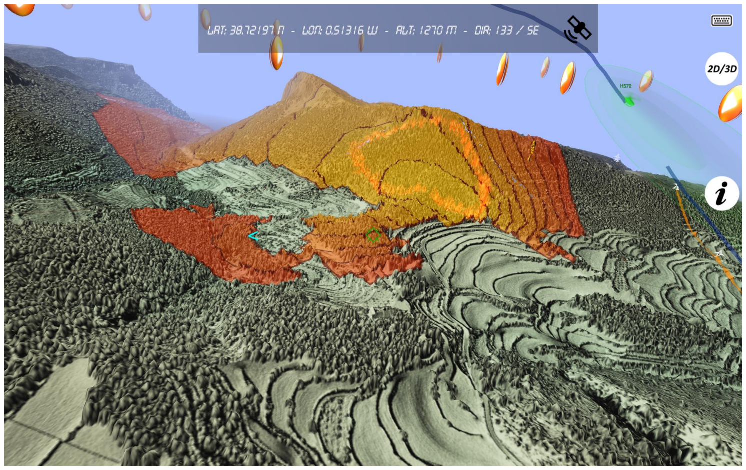

Fig. 5. Fire simulation

represents the current situation of the fire. From this central ring, we can see different black rings which represent, in step of 12 hours, the expected evolution of the fire scenario according with different fire models.

Due to the fact that most project partners had no prior knowledge about Ice, we developed an ad-hoc integration library. That library hides most of the details of the publication and subscription operations and it may create event channels when required. It also provides a specific signature because IceStorm is type agnostic. The library has been implemented in the Java, C# and Python, the languages the projects partners requested.

IceStorm provides another valuable feature, unusual in other middlewares; each peer (publisher or subscriber) may use a different transport protocol to communicate to the broker. Even a single entity that behaves as a subscriber and publisher at the same time, may use a different protocol per role. The supported protocols are UDP, TCP and SSL, but these are enough for the usual use cases, that is, reliable/non-reliable and secure transports, being the latter very important when the infrastructure requires intermediate third party networks.

5.1 Native sensor integration

Most modules may work independently. The middleware integration library provides bindings to easily encapsulate results to other modules and there de-encapsulate them to be used as input information. There is a clear boundary among modules inner implementation and the events channels. That boundary is responsible to convert and adapt formats, data types, name conventions, etc. However this adaption introduces some overhead regarding increased latency and message size.

For most of modules this issue is not relevant but in the case of the sensor network it implies some interesting issues. The described deployment entails a bridge device that receive messages from the sensors nodes (in a proprietary protocol) and convert them to events (object invocations in the IceStorm case).

It is advantageous to develop an end-to-end integration mechanism that allows all entities to send and receive conventional object invocations without intermediaries, avoiding thereby message adapting processes. In this sense, we are developing THEM (The Heterogeneous Embedded Middleware) that supports basic object oriented middleware capabilities and is suitable to be installed on very low footprint micro-controllers. THEM is based on specialized virtual machine installed in the sensor node. That machine interprets an ad-hoc generated program able to recognize and generate middleware protocol compliant messages. This way it is possible to set triggers that send conventional remote objects invocations when a physical event occurs.

6 Conclusions

This paper describes the integration process and resulting platform towards the coordination of the information flows gathered and generated during the wildfire extinction activities.

One of the main challenges that have been faced to accomplish this goal involves issues such as heterogeneity of platforms, underlying networks, event type and timing, or even the staff technical background that complicates the inter-work among all the parties involved. An additional limitation of traditional approaches is that data are highly dispersed all over the involved entities. Each of the involved entities generally generates or gathers information using their own formats, protocols, or semantics giving rise to the "information island" problem. Finally, one of the last challenge faced by this work is due to the limitations of the physical medium along with the high mobility of some of the entities supporting the communications. All these features belonging to the real world communication originate a low-reliability communication network, with highly variable delays, or lost of connectivity, among some.

The main contributions of this paper are intended to provide a solution to the aforementioned challenges. Regarding heterogeneity issues, the proposed solution resorts to a middleware platform that abstracts platforms and networks-dependent details by providing system-wide interfaces publicly available. In this sense, the use of a middleware assures that programmers count on a common API, made available as a library, whereas the use of the middleware communication channels forces a common message format that enables protocol-level interoperability.

One of the main novelties of this work is related to how the "information island" problem has been resolved. The proposed solution resorts to providing a common set of rules for event propagation. For example, apart from the timestamp attached to every event, if an event comes from a deployed device it also adds its GPS coordinates. It is important to highlight that combining all the collected events it is possible to reliably recreate, both in space and time, all the happenings that took during the extinction works. This information is extremely useful for forensic analysis intended to identify faults and propose future improvements.

The last contribution of this work is intended to relieve the involved entities from having to deal with the network delays and reliability issues. The proposed solution consists in enhancing the aforementioned library with transparent mechanism to temporally store the propagated events until the reliable link is available to be resent.

Acknowledgment

This research was supported by the Spanish Ministry of Science and Innovation and the Centre for the Development of Industrial Technology under the project PROMETEO (CEN-20101010), and by the Spanish Ministry of Economy and Competitiviness under the project DREAMS (TEC2011-28666-C04-03).

References

1. Z. Hui, *Study on Fire Fighting Information Integration System Architecture Based on SOA/Web Services*, 2011 Fourth International Symposium on Knowledge Acquisition and Modeling (KAM), pp.226-228, Oct. 2011.
2. S. Duan, K. Cheng, Q. Ge, C. Wen. *Wireless Intelligent Fire Fighting Systems Software Platform R & D*, International Conference on and 4th International Conference on Cyber, Physical and Social Computing Internet of Things (iThings/CPSCom) pp.94-99, Oct. 2011.
3. D. Yuan; X. Jin; X. Zhang, *Building a immersive environment for firefighting tactical training*, 9th IEEE International Conference on Networking, Sensing and Control (ICNSC), pp.307-309, April 2012.
4. Poggenpohl, F.; Guttinger, D., *Optimizing Task Allocation on Fire Fighting* 4th International Conference on Intelligent Networking and Collaborative Systems (INCoS), pp.497-502, Sept. 2012.
5. Pack, D.J.; Avanzato, R.; Ahlgren, D.J.; Verner, I.M., *Fire-fighting mobile robotics and interdisciplinary design-comparative perspectives* IEEE Transactions on Education, vol.47, no.3, pp.369,376, Aug. 2004.
6. I. Yoon; D. Kun Noh; D. Lee; Teguh, R.; Honma, T.; H. Shin, *Reliable Wildfire Monitoring with Sparsely Deployed Wireless Sensor Networks* IEEE 26th International Conference on Advanced Information Networking and Applications (AINA), pp.460-466, March 2012.
7. Pastor, E.; Sole, M.; Lopez, J.; Royo, P.; Barrado, C., *Helicopter-based wildfire monitoring system software architecture* IEEE Aerospace Conference, pp.1-18, March 2010.
8. Zapatero, M.; Castro, R.; Wainer, G.; Houssein, M., *Architecture for integrated modeling, simulation and visualization of environmental systems using GIS and Cell-DEVS* Proceedings of the Winter Simulation Conference (WSC), pp.997-1009, Dec. 2011.
9. Kontoes, C.; Keramitsoglou, I.; Papoutsis, I.; Herekakis, T.; Michail, D.; Xofis, P.; Koubarakis, M.; Kyzirakos, K.; Karpathiotakis, M.; Nikolaou, C.; Sioutis, M.; Garbis, G.; Vassos, S.; Manegold, S.; Kersten, M.; Pirk, H.; Ivanova, M., *Operational*

wildfire monitoring and disaster management support using state-of-the-art EO and Information Technologies, Second International Workshop on Earth Observation and Remote Sensing Applications (EORSA), pp.196-200, June 2012.
10. Michi Henning and Mark Spruiell, *Distributed Programming with Ice* Revision 3.4, June 2010, ZeroC Inc.
11. Dynastream Innovations, Inc., *ANT Message Protocol and Usage*, Revision 5.0, January 2013.
12. ZigBee Standards Organization, *ZigBee Specification (document 053474r17)*, January 2008.