

Research Article

Self-Organizing Architecture for Information Fusion in Distributed Sensor Networks

Javier Bajo,¹ Juan F. De Paz,² Gabriel Villarrubia,² and Juan M. Corchado²

¹Department of Artificial Intelligence, Polytechnic University of Madrid, Campus Montegancedo s/n, Boadilla del Monte, 28660 Madrid, Spain

²Faculty of Science, University of Salamanca, Plaza de la Merced s/n, 37002 Salamanca, Spain

Correspondence should be addressed to Javier Bajo; jbajo@fi.upm.es

Received 15 December 2014; Revised 1 March 2015; Accepted 7 March 2015

Academic Editor: Feng Hong

Copyright © 2015 Javier Bajo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The management of heterogeneous distributed sensor networks requires new solutions that can address the problem of automatically fusing the information coming from different sources in an efficient and effective manner. In the literature it is possible to find different types of data fusion and information fusion techniques in use today, but it is still a challenge to obtain systems that allow the automation or semiautomation of information processing and fusion. In this paper, we present a multiagent system that based on the organizational theory proposes a new model to automatically process and fuse information in heterogeneous distributed sensor networks. The proposed architecture is applied to a case study for indoor location where information is taken from different heterogeneous sensors.

1. Introduction

During the last decades, sensor networks have become increasingly relevant and nowadays are present in practically all sectors of our society [1]. Their great capacity to acquire data and act on the environment may facilitate the construction of smart environments, allowing a detailed and flexible analysis of processes that occur in the environment and the services that can be provided to the users.

Despite the many advances that exist today, the interconnection of sensor networks is challenging due to the complexity of interaction with the different existing technologies and communication protocols, each of which has certain requirements [2, 3]. Additionally, it is a challenge to efficiently manage information generated by sensors. This may be complicated because the data can be inconsistent due to different reasons, highlighting (i) the deployment over open environments which lack the ability to integrate sensor technologies (they are highly complex and heterogeneous) or (ii) the difficulty to merge the information from such sensor networks in a simple and efficient manner. Thus, at present there is a growing need for versatile open platforms capable of integrating different sensing technology over wireless technologies,

fusing data sets from heterogeneous sources, and intelligently managing the generated information. Nowadays there are already some architectures or frameworks that allow the interconnection of sensors, both in academia [2, 4, 5] and in industry [6–9]. However, the reality is that existing platforms are designed for a specific end, using a specific technology stack for each deployment and offering a very specific set of services whose functionality is very limited [10]. Therefore, current systems are limited by the preinstallation of infrastructure, and integrators have to face the decision of choosing between other technologies or adapting their existing systems and infrastructure [11]. It is also difficult for integrators to combine the information obtained from heterogeneous wireless sensor networks [12–14], since there are no adequate tools to do so.

In this paper, we present an open platform that will facilitate the integration and management of sensor networks and provide services based on the information provided by mentioned sensor networks. One of the main novelties of the platform is its open character, allowing the addition of networks built using different existing technologies as well as new technologies or sensor networks as well as the integration of the data in a cloud computing environment. Open systems

TABLE 1: Existing approaches for the management of WSNs.

Architecture	Main characteristics	Weaknesses
MANNA [42]	Proposing the functional information and physical management architectures that take into account specific characteristics of this type of network	No self-organizing No information fusion
SPRING [5, 43]	Sensor grid architecture, called the scalable proxy-based architecture for sensor grid	Specifically designed for sensor grid No information fusion
H-WSNMS [44]	Facilitating management functions for specific WSN applications from the individual WSN platforms	No self-organizing No information fusion
MARWIS [45]	Management of architecture for heterogeneous wireless sensor networks (WSNs). It supports common management tasks such as monitoring, (re)configuration, and updating program code in a WSN	No self-organizing No information fusion
Yu et al. [46]	Lightweight middleware system that supports wireless sensor networks (WSNs) to handle real-time network management using a hierarchical framework	No information fusion
G-Sense [47]	Integrating mobile and static wireless sensor networks in support of location-based services, participatory sensing, and human-centric sensing applications	No self-organizing No information fusion

exist in dynamic operative environments where new components are integrated or existing components leave the system continually and where even the operating conditions can change unpredictably. Open systems are characterized by the heterogeneity of their participants, limited trust, individual goals in conflict, and a great probability of disagreement with the specifications [15]. The proposed platform combines virtual organizations of multiagent systems [15–17] with information fusion algorithms [8, 12, 13], which provides great adaptability to changes in the controlled environment and a more efficient usage of the technologies available in that environment. In addition, we propose to integrate the virtual organization within cloud computing environments, providing ubiquitous computation and communication mechanisms, and an adequate environment for the application of intelligent algorithms, which will allow the extraction of knowledge from great volumes of data, as well as the customization of the services offered to the end user. These mechanisms differ from the existing security alternatives, which are based on firewall-like models and do not enable dynamic adaptation. As of today, a platform like this has not been developed, that is, a platform with an open character, able to incorporate information fusion algorithms, capable of integration with cloud technology [18] built upon the base of social computation.

This paper proposes a level-based information fusion architecture for distributed sensor networks. This model will be able to obtain information provided by the sensor networks and to be integrated with virtual organizations of agents. The model proposes the use of signal filtering algorithms, normalization services, and other signal processing services at a basic level. It also proposes new algorithms for information fusion at higher levels that could be integrated with intelligent agents, following processes similar to the mixture of experts [19], which will obtain the data coming from

heterogeneous sources and merge it according to the preferences of the environment. Each expert will be implemented by an intelligent agent, making it easy to dynamically add it to the architecture. Each mixture of experts will also be implemented by an intelligent agent. These systems will incorporate probabilistic models, Bayesian networks, fuzzy systems, and image recognition algorithms to process information from sensors. Also, the system will incorporate agents to process and automatically generate efficient information fusion processes using statistical analysis of workflows based statistics. The information fusion will incorporate neural networks, Bayesian models, and linear programming.

This paper is organized as follows: Section 2 revises the related work, Section 3 describes the proposed architecture, Section 4 presents the case study in which the platform is applied, and finally Section 5 shows the results and conclusions obtained.

2. Related Work

Nowadays, there is a lack of open platforms that integrate new technologies of sensors and that are capable of efficiently fusing information from such networks and have capacities for self-organizing [18, 20, 21]. If we revise the related work, it is possible to find that traditionally WSNs have been developed without considering a management solution that can dynamically adapt to the changes that occur in the environment and the user needs. In Table 1, we present a review of the existing approaches for the management of WSNs.

As can be seen in Table 1 most of the existing approaches are designed for specific environments or specific purposes and none of them combine incorporate capacities for self-organization and open integration. Besides, the possibility of using information fusion techniques is not explicitly defined as a capacity of the architecture. Although significant progress has been made in the development of architectures to manage

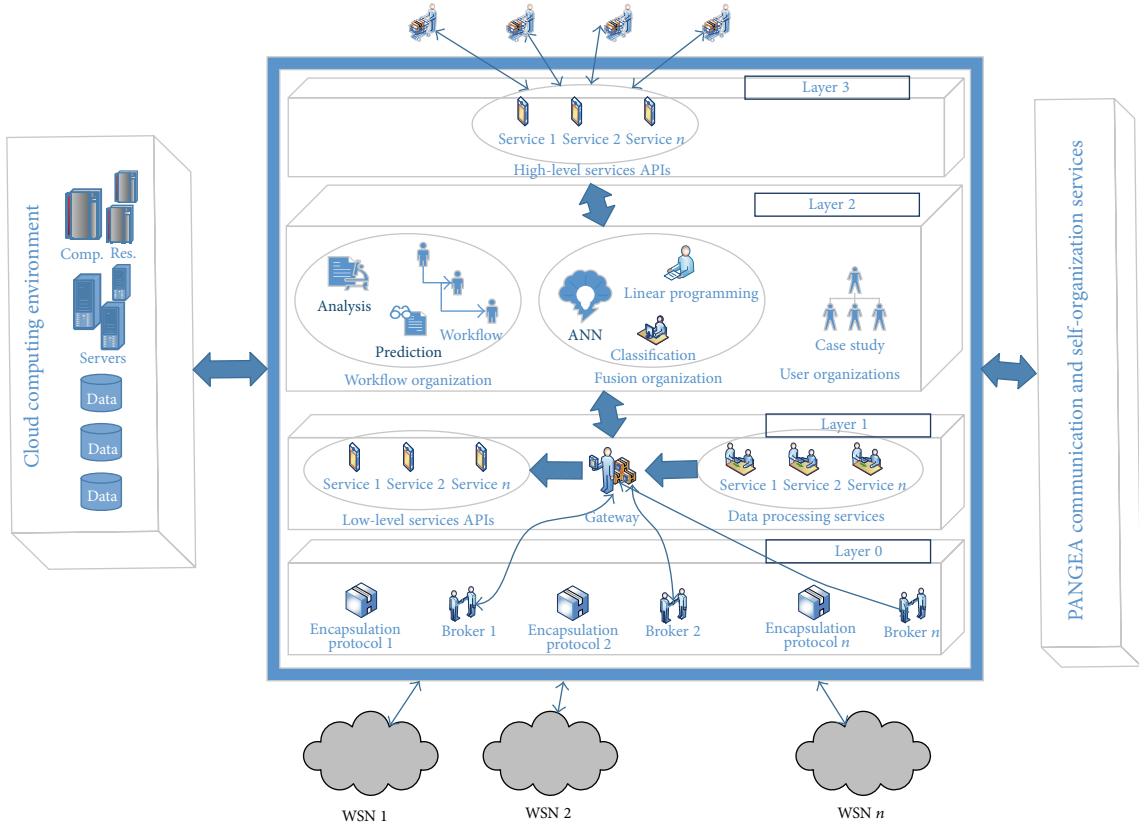


FIGURE 1: Data fusion architecture.

wireless sensor networks, at present there is not a single open platform that efficiently integrates heterogeneous wireless sensor networks and provides intelligent information fusion techniques and intelligent services. The use of multiagent systems with capabilities for self-organization and dynamic management of information fusion workflows can notably help to improve the management of heterogeneous sensor networks.

Information fusion is applied to classification problems or merging the solutions of different systems [22, 23]. However, it is not possible to find i solutions that allow the semiautomatic creation of workflows of information fusion that would reduce the development time integrating new components. Some existing classifiers include bagging and boosting [23, 24] techniques, which allow merging the outputs of several classifiers to improve the results of the individual ones. However, their results are not always satisfactory as we have proven in previous works [25, 26]. In order to find the desired behavior for our system, we can revise the structure of the mixtures of experts used in artificial intelligence, which merge the information based on the output provided by several experts [27]. This issue has been explored in several studies to improve the predictions processes [27, 28]. It is also common to find tools such as Spring Integration, Apache Camel, or even Weka, which can define workflows in data analysis. However, it would be of interest to obtain a new architecture that would learn from the information fusion workflows and generate new flows automatically or

semiautomatically, as previously stated [17, 29]. This led to the possibility of creating an open architecture with capacities for learning and self-organization making use of virtual organization of agents [15, 17, 29]. Virtual organizations of agents combine the multiagent paradigm and the theory of organization, allowing the design of artificial societies regulated by organizational aspects [30, 31]. In the next section, we describe our proposal, a multiagent architecture based on virtual organizations, specifically designed to manage distributed sensor networks in an organizational way.

3. Proposed Architecture

Our proposal consists of a multiagent architecture based on virtual organizations that integrates with an information fusion model. The architecture is designed on the basis of the previously developed PANGEA platform [31], a multiagent platform that facilitates the management of virtual organizations of agents, including self-organizing capacities. A complete description of PANGEA can be found in [31]. The new architecture is organized in 4 layers as can be seen in Figure 1 and is developed according to the classification of the JDL fusion model [31, 32].

- (i) *Layer 0: Integrate Sensing/Performance Technologies.* The platform will facilitate the integration of sensing technologies, regardless of their nature, and will provide an openness environment that allows dynamic

addition of new sensor systems and technologies. For this, the platform will provide data encapsulation mechanisms that standardize the information received from standard communication protocols such as WiFi and ZigBee. In this way, the platform will have sufficient dynamism to allow for the incorporation of emergent technologies. The incorporation of a new sensor network technology to the platform depends on communication technology used by the network. Layer 0 of the platform is a broker that defines communication with sensor networks of different nature (Wi-Fi, ZigBee, Bluetooth, etc.) and gets the raw data from sensor networks. This process of obtaining raw data from sensor networks is associated with Level 0, Data Assessment of the JDL model. This process of acquiring raw data from sensor networks is associated with Level 0, Data Assessment. The authors of this paper have conducted some preliminary research in which data from ZigBee and Wi-Fi sensors were obtained [10, 29–31]. The process of obtaining data requires studying the technologies and existing communication protocols for sensor networks and defining a new broker and adapters to provide the platform with openness. The main novelty of this layer is the ability to provide the platform and the upper layers with openness regarding the connection to sensor networks of different natures and thus ensure that upper layers of the architecture have access to information and are able to perform data fusion at different levels.

- (ii) *Layer 1: Low-Level Services.* The contextual information obtained from layer 0 is processed by a set of low-level services. After obtaining the raw data, a gateway [20] is defined through adapters that allow standardizing the information received. The data processing corresponds to Level 1, Object Assessment of the JDL model. In this first stage, the platform provides services such as filtering of signals, normalization services, or other treatment services at the basic level signals. These services are provided by the adapters and will be associated with algorithms that perform initial treatment of the data. Each of these services exposes an API to higher layers which allow interaction with each low-level service and thus with the sensing/performance underlying technologies.
- (iii) *Layer 2.* This layer includes Levels 2 to 4 of information fusion displayed on the JDL model. The platform is structured as a multiagent system based on virtual organizations. Each organization contains the roles required to facilitate an intelligent management of the information obtained from the lower levels of the architecture. The multiagent system incorporates agents specifically designed to interact with low-level services. In addition, we propose the design of intelligent agents specialized in information fusion. For this purpose, roles that allow merging information automatically through supervised learning and previous training are included. This procedure is similar to

the one used for the mixture of experts [32, 33]. The layer incorporates different statistical techniques based on linear programming [34], neural networks [32], and classifiers [34]. Thus, specialized agents in information fusion can combine different sensing technologies that provide heterogeneous data and provide more accurate information to the upper layer services. Besides providing information fusion techniques, this layer will provide an automatic generation of information fusion flows between different levels that will be described in detail in the following subsection.

- (iv) *Layer 3.* The top layer of the platform provides an innovative module that allows management and customization of services to end users, thanks to the capabilities of multiagent system which is deployed over the Cloud Computing environment. In this layer Levels 5 and 6 according to the JDL model are included. At these levels tasks associated with the man-machine interfaces are performed according to the characteristics of the user and facilitate decision making by the user.

3.1. Layer 2: Workflow and Fusion Organizations. In this paper, we will focus on layer 2 of the architecture, which defines different organizations of agents specialized in information fusion. Specifically we will focus on two organizations: workflow organization and fusion organization. The former aims to explore and find new workflows [17] that can analyze the data more optimally, while the latter creates a series of optimization techniques to optimize the process in terms of different variables. In addition to these two virtual organizations, the architecture will have its own organizations from the PANGEA architecture [31] and those of the case study as well.

3.1.1. Workflow Organization. This organization contains agents that play roles such as analysis workflow, prediction workflow, and workflow processing. The analysis workflow is responsible for analyzing a given workflow and for creating new workflows in an automatic way [21]. The workflows combine different information fusion techniques and algorithms integrated into the virtual organization for data processing for a specific case study. To carry out the analysis and prediction of the workflows, statistical techniques based on statistical models are applied to estimate the most appropriate execution flows [21]. The workflows obtained will be presented to the top layer, as a high-level API specific for each service, so that it is possible to use them in an easy and simple way.

The extraction of the relevant action is carried out applying different tests according to the characteristics of a given action. The chi-squared test is a nonparametric test that can be applied to analyze the independence of two variables X and Y ; the null and alternative hypotheses are H_0 : X and Y are independent and H_1 : X and Y are not independent. Applying this test is recommended ensuring that 80% of the expected frequencies of the contingency table have a value greater than 5. If the restriction is not met, it is possible to introduce

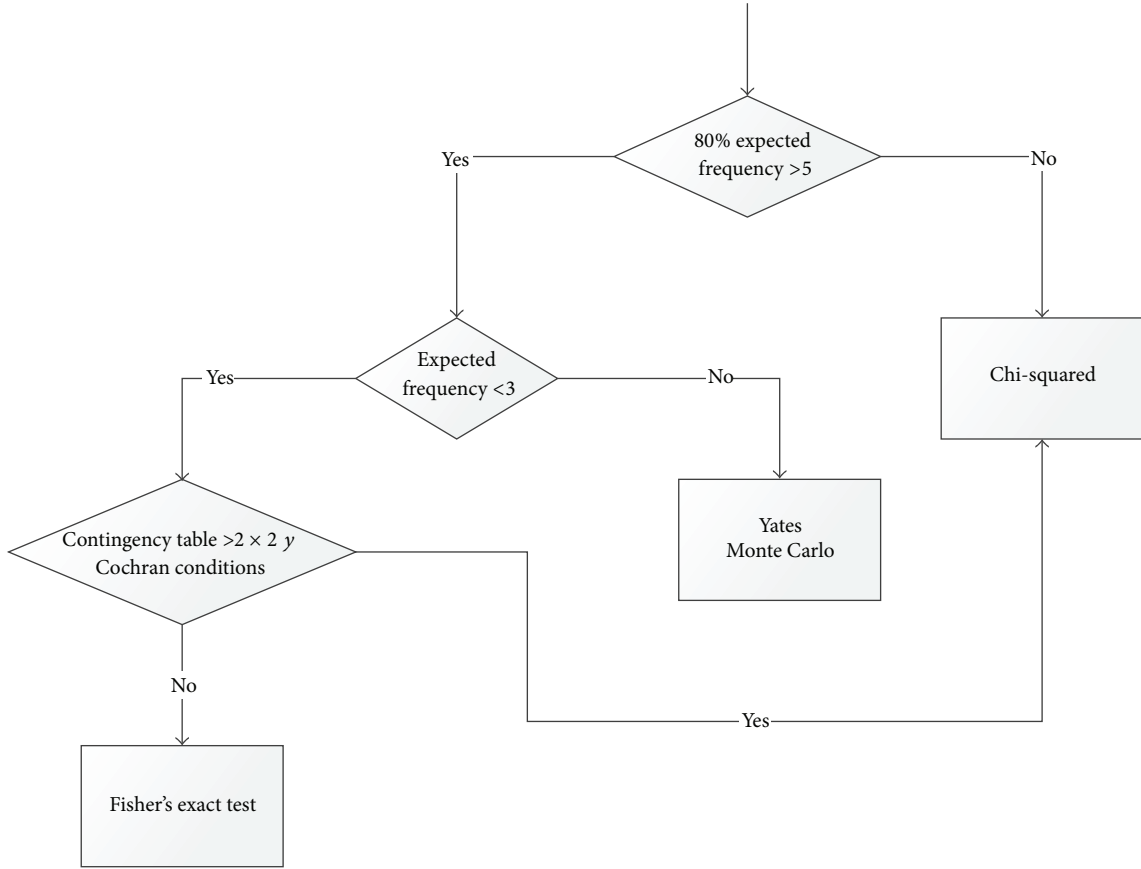


FIGURE 2: Flowchart to select the correct statistical test.

a correction by Yates [35] or Monte Carlo simulation [36] should be applied if the frequency is between 3 and 5. If the value is lower than 5 we should apply a Fisher's exact test [37]. The statistical test for the chi-square is defined by

$$\chi_{\text{exp}}^2 = \sum_{i=1}^r \sum_{j=1}^c \frac{(|o_{ij} - e_{ij}| - y)^2}{e_{ij}} \quad \text{with} \quad (1)$$

$$e_{ij} = \sum_{k=1}^c o_{ik} \cdot \sum_{k=1}^r o_{jk}$$

where o_{ij} is the value of the row i and column j of the contingency table and $y = 0$, where Yate's correction is introduced $y = 0.5$.

Once χ_{exp}^2 has been calculated, it is possible to define the significance level α defined by $\chi_{\text{exp}}^2 = \chi_{(n-1)(m-1); \alpha}^2$, which allow us to determine the importance of the variable.

Fisher's exact test is usually applied for 2×2 contingency tables but it can be applied to higher dimensions; for example, *R* software performs a chi-squared test if Cochran conditions are true and in other cases Fisher's exact test may be applied. Thus, the flowchart in Figure 2 represents the selection process for the statistical test to establish if a variable is relevant or not to the problem.

According to the α value, calculated in (1), it is possible to determine the degree of relevance for each of the possible actions, as shown in

$$W(B) = \sum_{j=1}^{|C|} (-1)^j \alpha \frac{n_j^S}{N^S} \frac{E_{\text{max}} - \bar{x}_j^S}{E_{\text{max}}}, \quad (2)$$

where B is the service that separates the set of data S in S_1, S_s , n_j^S is the number of elements in C_j ($j = 1$ nonefficient and $j = 2$ efficient), C are the classes in S , N^S is the elements in S , E_{max} is the maximum error in the workflows, and \bar{x}_j^S is the average error in S and class C_j .

Once the weights for each of the services are defined, it is possible to obtain a graph as shown in Figure 3 that captures the knowledge about the relevance of the actions. The graph contains information about the inputs and outputs of each of the services, which can limit the connectivity between the different nodes in the graph. To select a workflow, we apply Floyd's algorithm [22]. Those nodes that are connected to a fusion node are added to the workflow if the value obtained by the algorithm is higher than the previous one.

3.1.2. Fusion Organization. The fusion organization includes algorithms to select and combine the optimum technique(s) according to the characteristics of the service or information that will be fused. In this organization we define different

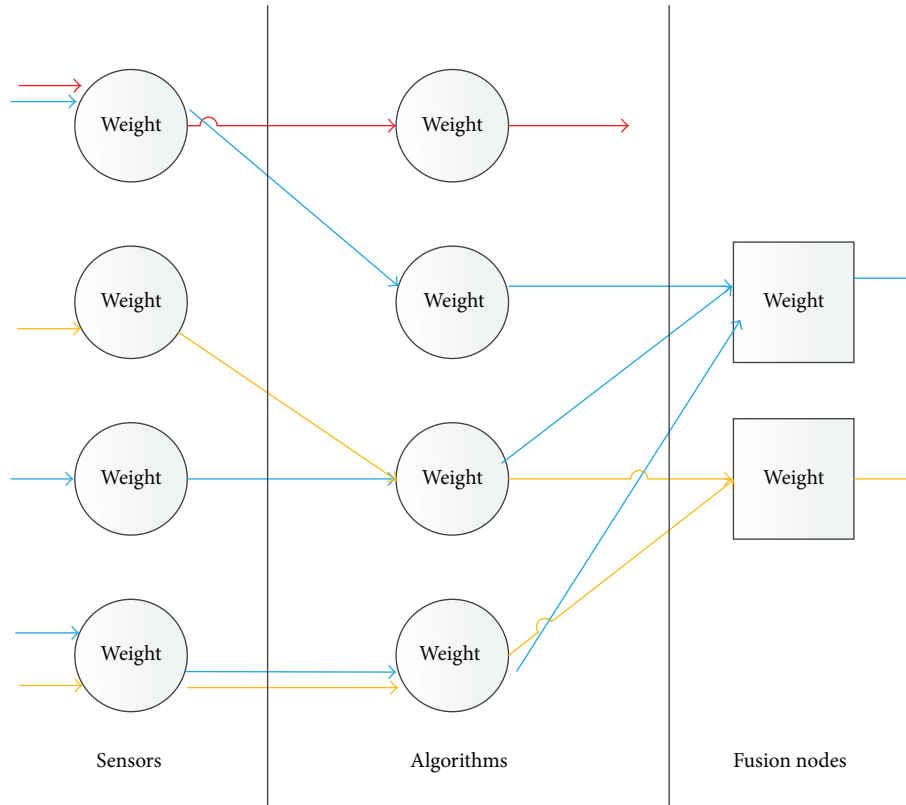


FIGURE 3: Actions workflow.

techniques for information fusion such as multilayer perceptron (MLP) [38] and Bayesian networks [39] that can work in a similar way than a mixture of experts, selecting the inputs that provide better results [25, 27]. These are supervised methods that require a previous training and it is necessary to have available a dataset for training the system. As an alternative to the supervised methods, we also incorporate a linear programming method to the fusion organization, based on the simplex algorithm [33].

In the next section, we present a case study where the proposed approach is evaluated. In the case study, we will focus on the information fusion capacities of the architecture.

4. Case Study

In this case study, we present the application of the proposed platform to an indoor location system that combines information coming from four heterogeneous sensor networks: WiFi system, BLE based system, accelerometers and compass, and cameras. The environment for testing the proposed architecture was located at the University of Salamanca, Spain, and contains a WiFi network, a BLE network, and a set of cameras and the users were provided with smartphones that gather information by means of the accelerometer and the compass. The architecture was deployed in the test environment making use of Wi-Fi agents, BLE agents, camera agents, user agents, workflow agents (analysis, prediction, and workflow agents), and fusion agents (ANN, classification, and linear

programming agents). The WiFi indoor location system is based on the use of fingerprints on a map of intensities. In a first stage, it is necessary to generate the maps of intensities. Once the maps are generated, the location of the users can be obtained by comparing the current level of intensities to those stored in the map of intensities. Thus, the first step is to obtain a representative number of samples of intensities that allow us to define a radio electric map of the environment. For example, Figure 4 corresponds to a high-level service of layer 3 of the architecture designed for mobile devices and presents a route carried out to obtain the initial fingerprints in a laboratory of the University of Salamanca. Each fingerprint is composed of different measurements about the levels of intensity of the WiFi signal and contains the MAC addresses and RSSI level of each of the hotspots detected at a given location.

Once the fingerprints have been obtained, we create a classifier that allows calculating the probability of belonging of a measurement to each of the available fingerprints based on a significance level and according to a normal distribution [39]. The reason for creating a classifier agent for each of the floors of the building is to avoid an excessive growing of the classifier agent and the corresponding computational cost for the Smartphone at the time of estimating the position of the user. To determine the specific floor where the user is located, we apply a RandomTree classification agent. Once the floor is determined, the classifier agent that was previously created for that floor is selected and the system can proceed

```

Input: FloorDetection, FloorFingerPrint, measures, PositionFingerPrint
Output:  $x, y, z$ 
// FloorDetection: RandomTree classifier
// FloorFingerPrint: Fingerprint in the different floor
// measures: BSSID1: RSSI1; ...; BSSIDN: RSSIN
// PositionFingerPrint: Matrix with X, Y, Z position for each fingerprint
// RandomTree classifier
 $f \leftarrow \text{determentFloor}(\text{FloorDetection}, \text{measure});$ 
//  $i$  is the fingerprint,  $i^f$  fingerprint  $i$  in floor  $f$ ,  $s$  is the
BSSID,  $z$  is a normal distribution,  $z_{\text{measures}}$  standardized value
 $z_{\alpha_{i^f s}} \leftarrow z_{\text{measures}}$ ;
// probably of belonging fingerprint  $i$  in floor  $f$ 
 $p_i^f \leftarrow \frac{\prod_s \alpha_{i^f s}}{\sum_{i^f} \prod_s \alpha_{i^f s}};$ 
 $\vec{p}^f \leftarrow \cup_i p_i^f;$ 
//  $x'$  coordinate weighted according to the probability
 $x' \leftarrow \text{PositionFingerPrint}^j \cdot \vec{X} * \vec{p}^f;$ 
 $y' \leftarrow \text{PositionFingerPrint}^j \cdot \vec{Y} * \vec{p}^f;$ 
// time series, weighted average
 $x \leftarrow (\alpha * x'(t) + (1 - \alpha)(\alpha * x(t-1) + (1 - \alpha)(\alpha * x(t-2) \dots)));$ 

```

ALGORITHM 1: User location algorithm.

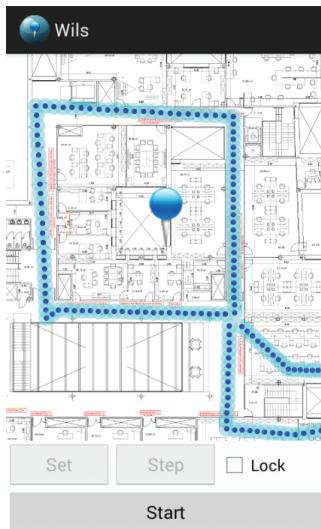


FIGURE 4: Screenshot of the WiFi indoor location service for smartphones.

to calculate the location of the user. This way, it is possible for us to reduce the time required to calculate the location of a given target and to have a more realistic approach when working with resource-constrained devices.

To calculate the user location, we use a Bayesian network integrated within a classifier agent that takes into account the probability for a given input of belonging to each of the previously scanned fingerprints. Taking into account the probability of belonging provided by the Bayesian network, we proceed to triangulate the location of the users in intermediate points of the route. Besides, to stabilize the position

of the users, we apply temporal series, which allow us to avoid continuous changes in the position of the users. The algorithm proposed to locate the users in the environment is shown in Algorithm 1.

The algorithm based on WiFi technology can be improved if the information from additional sensors is taken into account. More specifically, we use information coming from the accelerometer and compass of the Smartphone, the information provided by cameras, and the information provided by BLE tags. The accelerometer provides a way to define a step detection system that can be used together with the compass to locate the user. This system is available by default in modern smartphones that are used in this case study such as the Sony Xperia Z3, but for older devices, it is necessary to design a new algorithm based on thresholds [39] that is shown in Algorithm 2.

The main problem that presents a location system such as the one presented in this case study is the error reset. To correctly reset the error of the system it is required to determine the location in an accurate way. If this is not possible, the location will be provided based on the step detector and on the compass. To obtain the location of the user in a very accurate and exact manner, it is necessary to combine the WiFi system with complementary technology. In our case, we will use BLE tags and the camera of the mobile phone that were installed along the environment in the different floors of the building. Our first option was the use of BLE tags. We defined a threshold for the BLE signal and, when the RSSI level obtained for the signal is under the threshold and over the minimum detected value, we identify a situation where the user is close to a BLE tag.

The problem of the BLE tags is the variability of the signal when the user moves away from the tag. The levels of the

```

Input:  $x, y, z$ , upperThresholdActivated, lowerThresholdActivated
Output: upperThresholdActivated, lowerThresholdActivated, step
//  $x, y, z$  value of accelerometer with gravity
 $f \leftarrow \sqrt{x^2 + y^2 + z^2}$ ;
step  $\leftarrow$  false;
if  $f \geq$  upperThreshold && lowerThresholdActivated == 1 then
    upperThresholdActivated  $\leftarrow$  1;
    lowerThresholdActivated  $\leftarrow$  0;
    step  $\leftarrow$  true;
end
else if  $f \leq$  lowerThreshold && upperThresholdActivated == 1 then
    upperThresholdActivated  $\leftarrow$  0;
    lowerThresholdActivated  $\leftarrow$  1;
end

```

ALGORITHM 2: Step detection algorithm.

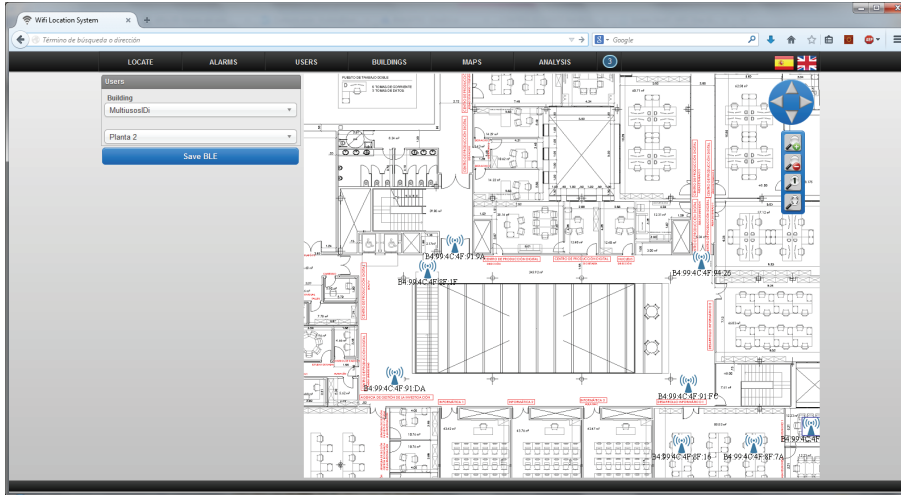


FIGURE 5: BLE tags location in the environment.

signal do not have a meaningful variation when the distance is up to 1.5 meters; thus, it is difficult to detect changes in the signal for instance when the distance is between 2.5 and 4 meters. In the case study, we distributed the BLE tags in each of the floors as shown in Figure 5. Figure 5 shows a high-level visualization service that is offered to the user in layer 3 of the architecture. There are locations where the BLE tags were installed by default such as elevators, stairs, corridors, or doors.

As an alternative to the use of BLE tags, we also use the camera of the mobile phone. The camera detects patterns that are used to locate the users in an accurate manner. The feature detection is carried out by means of the FAST algorithm [12] and the search of features is carried out through a kd-tree [40].

The information obtained from the different heterogeneous sensors is combined by means of a neural network integrated within a classifier agent. The MLP neural network acts as an information fusion mechanism and is designed taking into account the parameters presented in Table 2.

The set of low-level services that interact with the sensors and provide the necessary processing and normalizing capacities for the indoor location high-level service to the user is presented in Table 3.

5. Results and Conclusions

To evaluate results obtained testing the proposed architecture in the case study, we proceeded to calibrate three floors of the Multiusos R + D Building at the University of Salamanca, Spain. Each of the floors has an area of 53×54 meters. Algorithm 2 and Figure 5 show screenshots of the indoor localization service in the second floor of the building. The other two floors have a similar distribution of the space. The WiFi calibration process was conducted as shown in Algorithm 2, and we used a similar route for the rest of the floors. In the second floor of the building, we obtained a total of 2031 measurements distributed in 138 locations. That is, we obtained 138 fingerprints with 14 measurements for each fingerprint. In the case of the camera based system, we

TABLE 2: Input and output variables in the neural network.

Variable	I/O	Description
PositionRSSI	I	(x, y) coordinates x, y estimated. Corresponding to two inputs in the neural networks
StepBLE	I	Number of steps since max value of BLE detected
PositionBLE	I	(x, y) coordinates x, y estimated. Corresponding to two inputs in the neural networks
StepCam	I	Number of steps since last image
PositionCam	I	(x, y) coordinates x, y estimated. Corresponding to two inputs in the neural networks
PositionReal	O	(x, y) coordinates x, y . The true position of the user

TABLE 3: Low-level services available for the indoor location high-level service.

Service	Type	Description
WiFi	Sensor	Location system based on WiFi
AccelerometersCompass	Sensor	Location system based on accelerometers and compass
StepSensor	Sensor	Location system based on step sensor and compass
Camera	Sensor	Image capture
Step	Algorithm	Algorithm to detect step (Algorithm 2)
Image Recognition	Algorithm	Image recognition in the database
BayesWiFi	Algorithm	Location system based on WiFi [39]
Bayes	Fusion	Select the best predictor according to the position
MLP	Fusion	MLP to fuse the predictions of the other predictors
Linear programming	Fusion	Mixture of experts to fuse the information [33]

obtained 20 images per floor that correspond to the names of the rooms that are placed in the walls, close to the door of the room.

The training and identification of location areas were implemented through a training service, which is executed in the location server. This is due to the training process and the tests to analyze the functioning of the calibration process are carried out offline, since these are processed requiring dedicated computational resources. Furthermore, it is easier for the user to interact with the architecture to define the location areas in a simple manner as well as to monitor the calibration process. As can be seen in Figure 6, the user can define the location areas, in this case the second floor of the building is an area in our case study, and new polygon can be defined to create new areas.

Once the training data were obtained using the low-level services defined in layer 1 of the architecture, it was possible to establish the initial settings for the workflow and fusion organizations of agents. The initial test consisted of a calculation of the error for each of the individual location techniques used in the case study. In the case of the WiFi based algorithm, the calibration obtained during the training process was compared to the estimations obtained in a second route. In the case of the accelerometers and the camera, we obtained data in different routes in such a way that when a BLE tag or an image was detected, the difference between the estimation obtained and the real location of the BLE tag or the camera was computed as the error rate. For the rest of the points in the route, the error was computed for each of the steps of the user.

TABLE 4: Average with the different sensors.

Service	Average error
RSSI	1.81 m
Accelerometers	1.43 m
Camera	3.15 m

Table 4 shows the average error rate obtained for all the floors in the building for each of the individual location services defined in the architecture. The routes correspond to each of the lines presented in Figure 6 taking also into account crosses and changes of direction. Nine possible routes were defined for the second floor of the building. The accomplishment of the routes allows the evaluation of the image processing location technique and gathering data to evaluate the functioning of the WiFi location system.

The WiFi location service was tested using different classifiers. Table 5 shows the results obtained for the WiFi location service. As can be seen in Table 5, the WiFi location algorithm proposed in the architecture notably improved the performance of existing classifiers. The classification processes were executed on a i7 3612QM, 8 GB RAM server.

After testing the individual location services, we evaluated the self-organizing capacities for the architecture. More specifically, we evaluated the fusion organization (fusion location service) and workflow organization defined for the case study. In the case of the fusion organization, taking into account the definitions given in Section 3.1.2, we chose a MLP

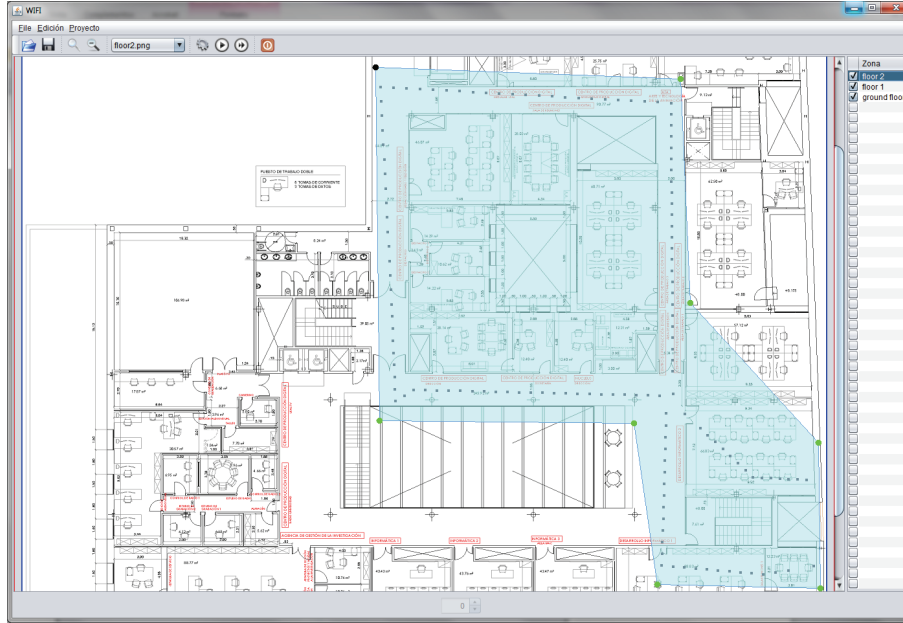


FIGURE 6: WiFi calibration service.

TABLE 5: Average error according to the different classifiers.

Average error (meters)	Execution time (ms)	Classifier
1.8123243	0.18233	Proposal
1.94445322	1.891679	BayesNet
2.42417577	125.890205	SimpleLogistic
2.05590806	14.889709	SMO
2.12398567	3.397341	IBk
2.5081823	0.351551	AttributeSelectedClassifier
2.38728749	1.13934	Bagging
2.44507039	10.972919	ClassificationViaRegression
2.30908468	0.160512	FilteredClassifier
2.12246602	17.963564	LogitBoost
3.43933531	3.988676	JRip
2.03425342	2.162482	PART
2.15537535	0.213195	J48
2.84376125	0.146233	REPTree

technique incorporated inside an agent at the fusion organization level to merge the location algorithms, as the inputs for the MLP are continuous. The self-adaptation capacities of the organization are mainly given by the MLP agent, as it automatically rescales the input variables to take values into the interval $[0.1, 0.9]$, and when there is no data available from a sensor, the value is -1 . Besides, the MLP agent was trained taking into account the inputs for each of the sensors and with a value of -1 for the rest of the inputs. The hidden layer of the MLP is designed with $2n + 1$ [41] neurons, where n is the number of neurons in the input layer. The output layer is composed of three neurons that correspond with the coordinates

(x, y, z) . In addition to the MLP, the linear programming method defined in [34] was also implemented in the fusion organization in order to schedule the different actions to be carried out.

In the case study, the workflow organization was trained to manage twenty flows that were applied according to the techniques shown in Table 2. The workflow organization allows us to dynamically define different possible workflows for the organization, which can be defined and used in real-time in the virtual organization. Thus, it is possible to dynamically make decisions in the workflow organization to provide the most efficient workflows for the problem at a given moment. Algorithm 2 shows three of these flows. The efficiency of each action was estimated according to the efficiency of each flow. The best flow was determined by applying the process presented in Section 3.1 and was estimated by the system. The error obtained was 1.43 meters, which is lower than the error obtained by the individual location techniques and lower than the error obtained using previous workflows. The workflow ranked with a lower error rate (1.57 meters) is the one shown in blue in Figure 7.

The system provides self-organizing capacities, as it can automatically adjust workflows and merge information from different low-level services to obtain a more precise location. The neural network can perform this task efficiently, improving the precision of other systems such as RSSI signal levels. The basic reason why the proposed approach improves previous results is that MLP selects the technique with a lower error rate taking into account the training data. For example, the WiFi location system does not work well in open spaces such as the area shown in Figure 8(a); thus, in these areas the location was obtained by means of the BLE-based location service. In the corridors in Figure 8(b), the accuracy of the location system was very high and if no BLE tags were

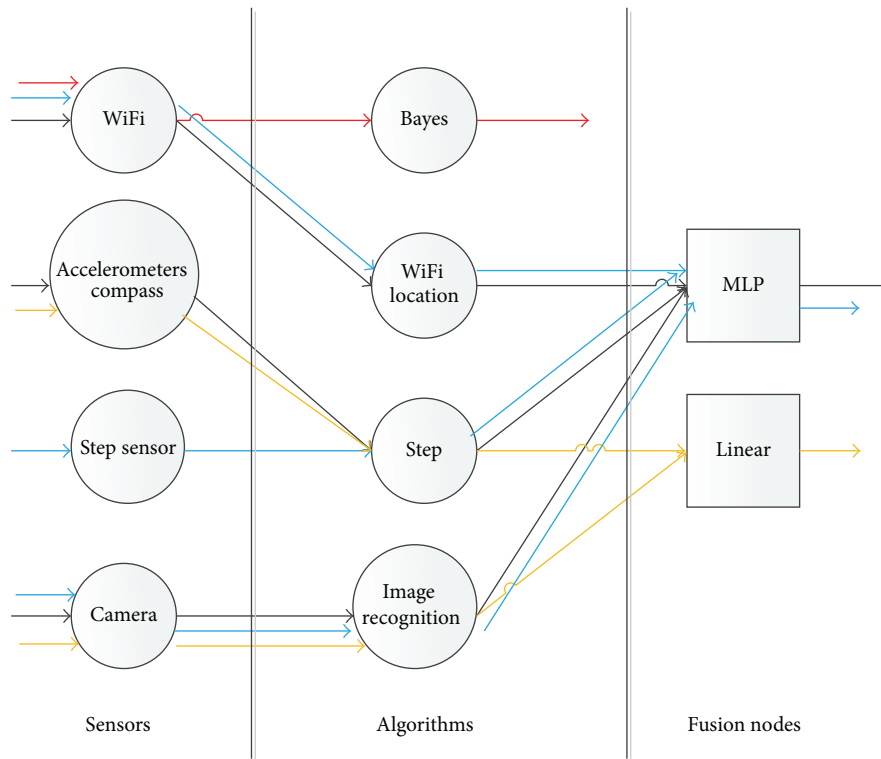


FIGURE 7: Execution flows in the system.

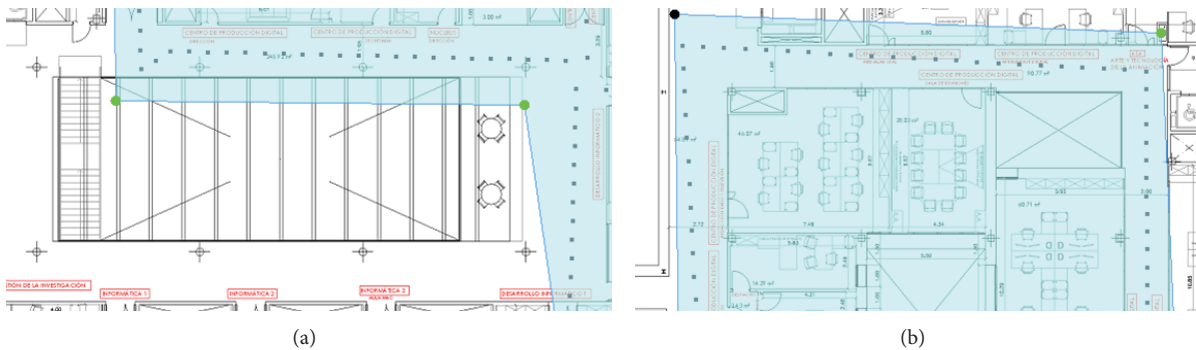


FIGURE 8: (a) Open space in second floor and 2 (b) corridors with no BLE tags.

available, the location was carried out via WiFi. The camera-based location systems were not used most of the times, given the difficulties for obtaining a natural way of using the camera of the mobile device to detect the patterns in the doors of the rooms.

The camera-based system processes the images and detects patterns that correspond to the names of the rooms. The performance of this system was not good mainly due to the problems to detect the pattern with the camera of the smartphone, as it is difficult for the user to use the camera and detect the patterns while walking. In the case of the BLE based location service, the performance can be considered as good, although it is possible to have problems when the user has an erratic behaviour and moves and changes his orientation continuously or agitates the smartphone. Regarding the sensors

of the smartphone, in the current service, the user orientation is determined by means of the position of the smartphone and taking into account that the user holds the mobile looking at the screen. If this position changes, or if the user moves in a different way compared to the usual one, the behaviour of the system can be erroneous. To determine the user orientation it is possible to take into account the linear accelerometer sensor of the smartphone, which together with the compass allow us to detect the direction of the movement of the user. This option will be evaluated in future works. The WiFi location system is the more stable, but the intensity maps have to be continuously updated.

From the point of view of the organizational architecture with information fusion capacities, we have proposed an adaptive approach that calculates the workflow with the

minimum cost and optimum error rate. It was necessary to include negative weights in the connections of the graph to avoid that the fusion nodes can add all the services that are connected to them. In this way, it was possible to reduce the number of connections for the fusion nodes. As an alternative to the workflow selection approach presented in this paper, it is possible to consider another gain function, as the Gini index or the Bayes probabilities. In this sense, there exist varieties of methods that can be incorporate to the architecture in order to fuse information in a dynamic way. In this paper, we have presented an innovative algorithm that proposes an information fusion technique depending on the variables in the environment: for continuous variable the MLP is selected, for nominal variables the classifier is the option chosen, and in those cases where a supervised learning technique is not possible, the system proposes a linear programming solution.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work has been supported by the Spanish Government through the project iHAS (Grant TIN2012-36586-C01/C02/C03) and FEDER funds.

References

- [1] L. Haibo and Z. Fang, "Design and implementation of wireless sensor network management systems based on WEBGIS," *Journal of Theoretical and Applied Information Technology*, vol. 49, no. 2, pp. 792–797, 2013.
- [2] S. V. Patel and K. Pandey, "Design of SOA based framework for collaborative cloud computing in wireless sensor networks," *International Journal of Grid and High Performance Computing*, vol. 2, no. 3, pp. 60–73, 2010.
- [3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–105, 2002.
- [4] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, "MQTT-S—a publish/subscribe protocol for wireless sensor networks," in *Proceedings of the 3rd IEEE/Create-Net International Conference on Communication System Software and Middleware (COM-SWARE '08)*, pp. 791–798, January 2008.
- [5] H. B. Lim, M. Iqbal, W. Wang, and Y. Yao, "The national weather sensor grid: a large-scale cyber-sensor infrastructure for environmental monitoring," *International Journal of Sensor Networks*, vol. 7, no. 1-2, pp. 19–36, 2010.
- [6] V. C. Gungor and G. P. Hancke, "Industrial wireless sensor networks: challenges, design principles, and technical approaches," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 4258–4265, 2009.
- [7] V. C. Gungor, B. Lu, and G. P. Hancke, "Opportunities and challenges of wireless sensor networks in smart grid," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 10, pp. 3557–3564, 2010.
- [8] G. Pavlin, P. de Oude, M. Maris, J. Nunnink, and T. Hood, "A multi-agent systems approach to distributed bayesian information fusion," *Information Fusion*, vol. 11, no. 3, pp. 267–282, 2010.
- [9] Z. Wang, L. Wang, A. I. Dounis, and R. Yang, "Multi-agent control system with information fusion based comfort model for smart buildings," *Applied Energy*, vol. 99, pp. 247–254, 2012.
- [10] D. I. Tapia, A. Abraham, J. M. Corchado, and R. S. Alonso, "Agents and ambient intelligence: case studies," *Journal of Ambient Intelligence and Humanized Computing*, vol. 1, no. 2, pp. 85–93, 2010.
- [11] D. I. Tapia, R. S. Alonso, Ó. Garcia, J. M. Corchado, and J. Bajo, "Wireless sensor networks, real-time locating systems and multi-agent systems: the perfect team," in *Proceedings of the 16th International Conference of Information Fusion (FUSION '13)*, pp. 2177–2184, July 2013.
- [12] N. Steinberg, C. Bowman, and F. White, "Revisions to the JDL data fusion model," in *Proceedings of the NATO/IRIS Conference*, Quebec, Canada, October 1998.
- [13] E. Blasch, "Level 5 (user refinement) issues supporting information fusion management," in *Proceedings of the 9th International Conference on Information Fusion (FUSION '06)*, pp. 1–8, July 2006.
- [14] N. Xiong and P. Svensson, "Multi-sensor management for information fusion: issues and approaches," *Information Fusion*, vol. 3, no. 2, pp. 163–186, 2002.
- [15] E. Argente, V. Julian, and V. Botti, "Multi-agent system development based on organizations," *Electronic Notes in Theoretical Computer Science*, vol. 150, no. 3, pp. 55–71, 2006.
- [16] E. Argente, V. Botti, C. Carrascosa, A. Giret, V. Julian, and M. Rebollo, "An abstract architecture for virtual organizations: the THOMAS approach," *Knowledge and Information Systems*, vol. 29, no. 2, pp. 379–403, 2011.
- [17] J. F. de Paz, J. Bajo, V. F. López, and J. M. Corchado, "Biomedical organizations: an intelligent dynamic architecture for KDD," *Information Sciences*, vol. 224, no. 1, pp. 49–61, 2013.
- [18] J. O. Gutierrez-Garcia and K.-M. Sim, "Self-organizing agents for service composition in cloud computing," in *Proceedings of the 2nd IEEE International Conference on Cloud Computing Technology and Science (CloudCom '10)*, pp. 59–66, December 2010.
- [19] M. H. Nguyen, H. A. Abbass, and R. I. Mckay, "A novel mixture of experts model based on cooperative coevolution," *Neurocomputing*, vol. 70, no. 1-3, pp. 155–163, 2006.
- [20] X. He, Y. Ma, and M. Mizutani, "Efficient multiple gateway system for WSN management in BEMS," in *Proceedings of the 9th International Conference on Networked Sensing Systems (INSS '12)*, pp. 1–2, IEEE, Antwerp, Belgium, June 2012.
- [21] D. Dragomir, A. Voinescu, A. Draghici, and N. Tapus, "WSN management in a multi-user secure context," in *Proceedings of the 11th Roedunet International Conference on Networking in Education and Research (RoEduNet '13)*, pp. 1–4, January 2013.
- [22] S. Hougardy, "The Floyd-Warshall algorithm on graphs with negative cycles," *Information Processing Letters*, vol. 110, no. 8-9, pp. 279–281, 2010.
- [23] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.
- [24] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proceedings of the 13th International Conference on Machine Learning*, pp. 148–156, Bari, Italy, July 1996.
- [25] C. I. Pinzón, J. F. De Paz, Á. Herrero, E. Corchado, J. Bajo, and J. M. Corchado, "IdMAS-SQL: Intrusion Detection Based on

- MAS to Detect and Block SQL injection through data mining,” *Information Sciences*, vol. 231, pp. 15–31, 2013.
- [26] C. I. Pinzón, J. F. de Paz, M. Navarro, J. Bajo, V. Julián, and J. M. Corchado, “Real-time CBR-agent with a mixture of experts in the reuse stage to classify and detect DoS attacks,” *Applied Soft Computing Journal*, vol. 11, no. 7, pp. 4384–4398, 2011.
- [27] J. M. Corchado, J. F. De Paz, S. Rodríguez, and J. Bajo, “Model of experts for decision support in the diagnosis of leukemia patients,” *Artificial Intelligence in Medicine*, vol. 46, no. 3, pp. 179–200, 2009.
- [28] J. F. de Paz, J. Bajo, A. González, S. Rodríguez, and J. M. Corchado, “Combining case-based reasoning systems and support vector regression to evaluate the atmosphere-ocean interaction,” *Knowledge and Information Systems*, vol. 30, no. 1, pp. 155–177, 2012.
- [29] G. Villarrubia, J. Bajo, J. F. de Paz, and J. M. Corchado, “Open multi-agent architecture for information fusion,” in *Proceedings of the 17th International Conference on Information Fusion (FUSION '14)*, pp. 1–8, Salamanca, Spain, July 2014.
- [30] S. Rodriguez, V. Julián, J. Bajo, C. Carrascosa, V. Botti, and J. M. Corchado, “Agent-based virtual organization architecture,” *Engineering Applications of Artificial Intelligence*, vol. 24, no. 5, pp. 895–910, 2011.
- [31] C. Zato, G. Villarrubia, A. Sánchez et al., “PANGEA—platform for automatic construction of organizations of intelligent agents,” in *Distributed Computing and Artificial Intelligence: 9th International Conference*, vol. 151 of *Advances in Intelligent and Soft Computing*, pp. 229–239, Springer, Berlin, Germany, 2012.
- [32] J. Bajo, M. L. Borrajo, J. F. de Paz, J. M. Corchado, and M. A. Pellicer, “A multi-agent system for web-based risk management in small and medium business,” *Expert Systems with Applications*, vol. 39, no. 8, pp. 6921–6931, 2012.
- [33] C. A. M. Lima, A. L. V. Coelho, and F. J. Von Zuben, “Hybridizing mixtures of experts with support vector machines: investigation into nonlinear dynamic systems identification,” *Information Sciences*, vol. 177, no. 10, pp. 2049–2074, 2007.
- [34] M.-P. Pere, S. Rodríguez, J. F. De Paz, R.-B. Ramon, M. P. Rubio, and J. Bajo, “Stereo video surveillance multi-agent system: new solutions for human motion analysis,” *Journal of Mathematical Imaging and Vision*, vol. 42, no. 2-3, pp. 176–195, 2012.
- [35] K. F. Wallis, “Chi-squared tests of interval and density forecasts, and the Bank of England’s fan charts,” *International Journal of Forecasting*, vol. 19, no. 2, pp. 165–175, 2003.
- [36] B. H. Baltagi, G. Bresson, and A. Pirotte, “Testing the fixed effects restrictions? A Monte Carlo study of Chamberlain’s Minimum Chi-Squared test,” *Statistics and Probability Letters*, vol. 79, no. 10, pp. 1358–1362, 2009.
- [37] F. Requena and N. M. Ciudad, “A major improvement to the Network Algorithm for Fisher’s Exact Test in $2 \times c$ contingency tables,” *Computational Statistics and Data Analysis*, vol. 51, no. 2, pp. 490–498, 2006.
- [38] J. F. de Paz, D. I. Tapia, R. S. Alonso, C. I. Pinzón, J. Bajo, and J. M. Corchado, “Mitigation of the ground reflection effect in real-time locating systems based on wireless sensor networks by using artificial neural networks,” *Knowledge and Information Systems*, vol. 34, no. 1, pp. 193–217, 2013.
- [39] G. Villarrubia, J. Bajo, J. F. de Paz, and J. M. Corchado, “Monitoring and detection platform to prevent anomalous situations in home care,” *Sensors*, vol. 14, no. 6, pp. 9900–9921, 2014.
- [40] Y.-S. Kang, J.-H. Nah, W.-C. Park, and S.-B. Yang, “Gkdtree: a group-based parallel update kd-tree for interactive ray tracing,” *Journal of Systems Architecture*, vol. 59, no. 3, pp. 166–175, 2013.
- [41] D. A. Sprecher, “A numerical implementation of Kolmogorov’s superpositions,” *Neural Networks*, vol. 9, no. 5, pp. 765–772, 1996.
- [42] L. B. Ruiz, J. M. Nogueira, and A. A. F. Loureiro, “MANNA: a management architecture for wireless sensor networks,” *IEEE Communications Magazine*, vol. 41, no. 2, pp. 116–125, 2003.
- [43] H. B. Lim, Y. M. Teo, P. Mukherjee, V. T. Lam, W. F. Wong, and S. See, “Sensor grid: integration of wireless sensor networks and the grid,” in *Proceedings of the IEEE Conference on Local Computer Networks—30th Anniversary (LCN '05)*, pp. 91–99, November 2005.
- [44] W. Zhao, Y. Liang, Q. Yu, and Y. Sui, “H-WSNMS: a web-based heterogeneous wireless sensor networks management system architecture,” in *Proceedings of the 12th International Conference on Network-Based Information Systems (NBIS '09)*, pp. 155–162, August 2009.
- [45] G. Wagenknecht, M. Anwander, T. Braun, T. Staub, J. Matheka, and S. Morgenthaler, “MARWIS: a management architecture for heterogeneous wireless sensor networks,” in *Wired/Wireless Internet Communications*, vol. 5031 of *Lecture Notes in Computer Science*, pp. 177–188, Springer, Berlin, Germany, 2008.
- [46] M. Yu, H. Mokhtar, and M. Merabti, “A self-organised middleware architecture for Wireless Sensor Network management,” *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 3, no. 3, pp. 135–145, 2008.
- [47] A. J. Perez, M. A. Labrador, and S. J. Barbeau, “G-Sense: a scalable architecture for global sensing and monitoring,” *IEEE Network*, vol. 24, no. 4, pp. 57–64, 2010.