Engineering Applications of Artificial Intelligence 87 (2020) 103243

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Multi-agent architecture for information retrieval and intelligent monitoring by UAVs in known environments affected by catastrophes ☆

D. Vallejo *, J.J. Castro-Schez, C. Glez-Morcillo, J. Albusac

School of Computer Science (University of Castilla-La Mancha) Paseo de la Universidad 4, 13071, Ciudad Real, Spain

ARTICLE INFO

Keywords:
Multi-agent architecture
Smart UAVs
Intelligent monitoring
Expert Systems
Crisis Management

ABSTRACT

The consequences of natural or man-made catastrophes can be devastating. To minimize its impact, it is crucial to carry out a rapid analysis of the affected environment in the moments after they occur, especially from the perspective of alert notification or crisis management. In this context, the use of UAVs, understood as the technological basis on which intelligent systems capable of providing support to rescue teams is built, has positively contributed to face this challenge. In this article the design of a multi-agent architecture which enables the deployment of systems made up of intelligent agents that can monitor environments affected by a catastrophe and provide support to human staff in the decision-making process is proposed. These environments, known in advance, are characterized through a set of points of interests that are critical from the point of view of aerial surveillance and monitoring. To conduct an intelligent information analysis, a formal model of normality analysis is employed, which makes possible the definition of surveillance components. These represent the knowledge bases of the agents responsible for monitoring environments. Likewise, the architecture envisages communication and cooperation mechanisms between the different agents, as the basis for fusing information to assess the overall level of risk of the monitored environment. A case study is presented in which the spread of toxic smoke in an industrial complex which has just suffered a hypothetical earthquake is monitored.

1. Introduction

The capacity to react in the moments immediately after a catastrophe, whether this is natural, such as an earthquake, or man-made, such as that caused by a nuclear reactor meltdown, is critical for minimizing the caused human and material damages. Unfortunately, in situations as extreme as these ones, it is highly complicated to tackle the chaos that usually arises. On occasions, this is due to the lack of information in such moments. Taking an earthquake that has affected an urban area as an example, with different types of buildings, and the difficulty associated with obtaining information about what is tracking place and which parts of the environment are in a critical state. In these situations, solutions based on independent robots which cooperate to obtain information about the environment and to analyze what is happening are especially important. If these are damaged in the rescue process, losses will only be economic ones. In recent years, research work linked to the use of independent robots used as a rescue tool in catastrophes and emergency situations has gained great importance, with one outstanding aspect of this being their independence in relation to human intervention and coordination (Murphy, 2004). Two more illustrative examples are shown by the intervention of independent robots in the collapse of the World Trade Center in the United States (Casper and Murphy, 2003) and in the Fukushima power station meltdown in Japan (Nagatani et al., 2013).

Although traditional robots represent a very good solution for preventing human staff putting themselves in danger when applying rescuing protocols, sometimes they cannot effectively operate due to the nature of the catastrophe. In this context, the use of unmanned aerial vehicles (UAVs), commonly known as drones, is ideal. Take, for example, the series of earthquakes which occurred in 2016 in central Italy with a death toll of 299 and substantial material damage. In similar situations, a mini-fleet of drones could be used as a first line of action to autonomously gather information in an environment with poor accessibility, quickly and accurately, thereby supplementing independent terrestrial robots and the human staff themselves. This approach could be based on the use of coordination and cooperation mechanisms, reducing the time needed to sweep the affected area and minimizing the damage the catastrophe may cause in the moments after it strikes.

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.engappai.2019.103243.

* Corresponding author.
E-mail address: David.Vallejo@uclm.es (D. Vallejo).
URL: http://www.esi.uclm.es/www/dvallejo (D. Vallejo).

https://doi.org/10.1016/j.engappai.2019.103243
Received 1 February 2019; Received in revised form 9 September 2019; Accepted 12 September 2019
Available online xxxx

Thanks to the flexibility and the rapid response provided by drones, it is no coincidence that they are currently in use in different domains, such as the military field, maps reconstruction, surveillance or goods transport, providing affordable solutions and a very high response time. In fact, a drone can be understood as a autonomous device that can collect information by means of a range of sensors (cameras, temperature sensors, presence detectors, etc.) quickly and in situations which it would be really complicated for other mobile devices. To be precise, designing systems based on independent drones, which can cooperate and coordinate with each other and with other agents, sharing information in order to reach an overall goal and to make decisions poses today a real challenge for the research community.

So, how can we assure that a set of drones work in cooperation with each other to solve the problem of information retrieval and intelligent analysis, quickly and efficiently, in environments affected by natural or man-made catastrophes? Then, as a supplementary question, how can we make the most of all current research in the field of intelligent surveillance to find out what is taking place in an environment?

Regarding information retrieval in the moments right after a catastrophe has occurred, a fleet of drones which suitably cooperate would provide that initial analysis of the affected area. In this way, the human agents could evaluate the situation and act in the most effective way possible. Apart from needing a cooperation layout, the system proposed herein must be scalable and independent from the number of available drones and adaptable to the monitoring needs of each situation. So, for example, a terrestrial robot may request a drone to act when it is not possible to access a certain point deemed critical from an information retrieval perspective.

From a computational point of view, both physical drones and terrestrial robots could be controlled by software agents (Wooldridge and Jennings, 1995) that intelligently cooperate to provide a solution to the previously stated problem. These agents would be capable of acting independently, obtaining and analyzing information about the environment, and communicating with each other, to make up a multi-agent system (Weiss, 1999) that acts as a whole in order to approach the posed challenge. It must be stressed that multi-agent systems adapt very well to problems in which complex situations are tackled where there are a range of heterogeneous information sources (Wooldridge, 2001). In particular, multi-agent systems have been extensively used in recent years to solve problems in which cooperation, planning, negotiation or distributed decision-making are essential to ensure key characteristics such as robustness, scalability and adaptability (Weiss, 1999).

To make possible the deployment of these multi-agent systems, in this paper a multi-agent architecture composed of different software agents is proposed. These agents play a series of defined roles to solve the problem of retrieving, analyzing and fusing information in catastrophic situations, and, ultimately, to provide support in the decision-making process. Outstanding among all the agents put forward are those known as information analysis agents, which are capable of analyzing what is happening in the environment thanks to the expert knowledge defined by means of a formal model of normality analysis (Albusac et al., 2009). This model, satisfactorily used in the field of intelligent surveillance of urban traffic environments (Albusac et al., 2010), is based on designing surveillance components, which combine both the knowledge base and the inference engine used to carry out this analysis. Moreover, the architecture considers the use of a series of communication mechanisms, such as the event channels and the blackboard architecture (Jennings et al., 1998), that provide the agents with flexibility when sharing information.

In the article a case study which contemplates monitoring the potential spread of toxic smoke in a scenario made up of an industrial complex and a nearby city is discussed. This example is used to show how the proposed architecture enables the deployment of a specific multi-agent system that carries out this monitoring task. Moreover, how to define a surveillance component, used to monitor the spread of toxic smoke, is described.

The remainder of the article is structured as follows. Section 2 compiles the main research works regarding the state of the art. Section 3 describes in depth the multi-agent architecture considered in this article, the reasons why it is necessary and its main contributions, stressing the roles of the different software agents designed and the definition of the integrated communication and cooperation mechanisms. Section 4 shows the main steps to be taken when deploying a multi-agent system from the architecture, regarding a case study in which the spread of toxic smoke is monitored, after a hypothetical earthquake, in a scenario made up of an industrial complex and a city. In Section 5, there is a definition of a surveillance component, aimed at monitoring the spread of toxic smoke, using the previously mentioned formal model of normality analysis (Albusac et al., 2009). Finally, Section 6 draws the obtained conclusions and outlines future lines of research.

2. Related work

2.1. UAVs and disaster management

The UAVs equipped with sensors which can remotely send information about an environment provide a wide range of possibilities in disaster-related situations. Here, the reader is referred to the study made by Adams and Friedland (2011) to obtain an overall perspective of the associated problem.

One of the situations which best represents a catastrophe is the earthquake. In this context, some authors such as Xu et al. (2014) consider developing a multi-drone system designed for obtaining information quickly in the moments after an earthquake. The architecture designed is made up of three items: (i) the multi-drone system itself, (ii) an earth station and (iii) an image processing system. In particular, the authors have paid special attention to making a very well-defined work flow and focus on obtaining high quality images in order to detect roads, receive information about the damage caused and for making an initial diagnostic of the situation. In the field of earthquakes, other authors have stressed planning the flights of the drones in order to make a stereoscopic reconstruction of the study area (Baiocchi et al., 2013). In this research, the tests carried out took place in the historic center of L'Aquila, a region that in 2009 was shaken by an earthquake in which there were hundreds of victims and considerable material damage. Another related study, albeit in the context of typhoons, is presented in Chou et al. (2010).

There is other interesting research which looks at solving more general problems which can be applied to the area of catastrophes. A representative example of this can be seen by reconstructing 3D environments from 2D images using drones, which sets two unrelated problems: (i) obtaining images by coverage path planning and (ii) reconstructing the scenario in 3D. In this setting, Torres et al. focus on obtaining images and design an algorithm in order to calculate a route for a drone which reduces battery consumption, by minimizing the number of turns the drone makes (Torres et al., 2016).

Another highly important area which involves managing catastrophes is the decision-making process (Power et al., 2015). Obviously, in this area coordination is fundamental. One possible approach is discussed in Fikar et al. (2016), where the authors put forward a system for backing decisions based on simulation and optimization in order to promote coordination between different rescue teams involved in an emergency. The layout under consideration makes use of an agent-based simulation, heuristics planning and Tabu searches in order to analyze the decision-making problem. The results are simulated in two regions close to the Danube river in Austria. Another interesting work is discussed by Zhao et al. (2019), where the authors address the cooperative decision-making problem for multi-target tracking when using systems composed of UAVs. An approach based on distributed multi-agent partially observable Markov decision processes, which is used to make tracking decisions.

Detecting the movement of people in situations such as earthquakes or floods is also essential, and, in recent years, a range of researchers have focused their work on using UAVs. In fact, some authors have directly focused on detecting victims (Andriluka et al., 2010). At present, a technique which is considerably common is using thermal cameras such as, for example, in Portmann et al. (2014), in which the authors propose a framework based on filtering particles in order to monitor people, using images taken by aerial devices. In the same context, in Gaszczak et al. (2011) a layout based on a range of Haar classifiers for automatically detecting vehicles is discussed.

Finally, there are publications whose results may be applied to the area under discussion in this research and which deal with more cross aspects of managing and controlling drones. Some representative examples of this can be seen in the independent control of drones (Kurnaz et al., 2010), the calculation of optimum routes (Sun et al., 2011), avoidance of obstacles and collisions between drones (Arokiasami et al., 2016) or cooperative searches (Beard and McLain, 2003; Ji et al., 2015).

2.2. Multi-robot systems

Nowadays, the term multi-robot implicitly involves refer systems composed of autonomous robots that are able to understand the environment where they live and act accordingly, maintaining internal models based on learning or inferring the knowledge required to perform tasks and adapt to the context (Ingrand and Ghallab, 2017). The evolution of this research area covers theoretical, computational and experimental aspects of multi-robot systems.

Multi-robot coordination is a relatively established line of research and one which we can directly make use of in this study, and, although the use of drones has been contemplated especially from an experimental point of view, it can be applied with a global outlook in which terrestrial robots and aerial drones work together (see Dias et al. (2006) and Yan et al. (2013) in order to obtain an overall perspective).

One coordination model that is particularly popular is the so-called emergent cooperation, used in the field of swarm robotics (Fukuda et al., 1988), where cooperative behavior on a group level arises between robots by means of interactions among them and with respect to the world, without them needing to explicitly work together. Although it is a simple solution, in the field of this research more direct cooperation aimed at trouble-shooting is required thanks to the skills robots and drones have. In this way a layout based on an intentional cooperation model is more desirable (Parker, 1998). With this model, robots cooperate explicitly and communicate based on the tasks they need to solve. It represents a layout closer to the real world in which robots do what humans wish them to do.

As regards the specific field of coordination with physical robots, some works which are based on the ALLIANCE architecture (Parker, 1998), reflect the international cooperation model, in order to manage how the robots which interact in order to complete their tasks behave (Arkin, 1998; Mataric, 1997).

On the other hand, coordinating a large number of robots is also a challenge that some authors explicitly address within the context of robot swarms. A possible approach consists in making use of a methodology that decomposes robot swarms into multiple groups in order to increase the versatility of the whole community of robots (Haghighi and Cheah, 2012). Thus, it is possible to deal with multiple levels of coordination in order to reach a common goal but reducing the complexity of coordinating multiple robots.

Multi-robot coordination should ideally deal with constraints that arise when deploying a team of robots in a physical environment where communication is restricted. Within this context, some authors propose solutions that are based on the use of periodic connectivity (Hollinger and Singh, 2012), which represents an aspect that should be considered in scenarios affected by catastrophes.

Additionally, another problem which merits special attention, and which is related to managing catastrophes and coordination is multi-robot exploration. In this respect, it is possible to consider a layout based on the premise that the objective of every robot is to minimize overall exploration time (Burgard et al., 2005), whilst simultaneously considering the cost of reaching a point of interest and how useful this is. Likewise, learning algorithms have also been used in dynamic and unknown environments in order to solve the multi-robot coordination problem. A representative example of this can be seen in Wang and de Silva (2008), where an approach based on using reinforcement learning and genetic algorithms is discussed in the context of an architecture of agents which take decisions in order to guide the robots to a point, avoiding obstacles. Moreover, there are approximations based on probability models, such as, for example in Claes et al. (2015), where the problem of coordinating a robot grid is tackled as well as choosing the best option for moving them.

Multi-robot patrolling (Portugal and Rocha, 2013a), usually carried out by a dynamic team of robots, is closely related to multi-robot exploration. Again, distributed strategies are gaining more and more relevance in order to increase system robustness and flexibility (Portugal and Rocha, 2013b), especially in environments where failures and communication errors might appear as a consequence of natural disasters or man-made catastrophes. Both exploration and patrolling rely on the data gathered from heterogeneous sensors, usually deployed on mobile robots. The process of fusing such data can be addressed in a decentralized way to perform cooperative perception (Capitán et al., 2011).

Multi-robot systems are inherently related to multi-agent systems, with the former adding the extra layer of complexity that derives from dealing with physical entities. The challenges faced are similar in a certain way, since they share the distributed nature of the solutions that are devised. Generally, centralized solutions are more efficient. So, why then should we be interested in distributed solutions which, normally, are more complex? There are a range of answers to this question, but, essentially, sometimes the problem itself is of a distributed nature. On other occasions, it is impossible to consider a centralized solution if data belong to organizations which need to keep them confidential. Other times, the necessary information for solving a problem is essentially distributed and lies in different systems which are large and complex and have various dimensions (Weiss, 1999): (i) the information is distributed geographically, (ii) there may be a high number of components, (iii) the number of concepts to be dealt with and the data associated with each of them may be enormous and, (iv) they may be very extensive in scope, covering a significant part of a determined field of work. From the point of view of standardization, it is worth mentioning the existence of initiatives to promote agent technology and standardization (Foundation for Intelligent Physical Agents FIPA, 2004).

At the same time, coordination becomes an essential aspect of multi-agent systems when facing problems in a distributed way. In fact, recent advances on distributed multi-agent coordination (Cao et al., 2012) show the impact they have on unmanned vehicles in terms of consensus, formation control, estimation, or optimization. Although cooperation-based approaches have traditionally drawn more attention, competition-based solutions also represent an alternative that can contribute to improve system performance.

As using independent drones is a recent area of research, there is not a large amount of important work which focuses on multi-agent architectures within the field of tackling catastrophic situations by means of UAVs or combining them with terrestrial robots. One recent example of this is SOIFRA (Service-Oriented Interoperable Framework for Robot Autonomy) (Arokiasami et al., 2016), considered from a multi-agent perspective for creating algorithms which are independent of the final platform, such as for detecting obstacles and preventing collisions, both for terrestrial robots and for UAVs. The agents designed maintain a BDI architecture (Rao and Georgeff, 1991) (Belief–Desire–Intention) and make use of JADE as a development and communication platform between agents (Bellifemine et al., 1999).

Another example of the general architecture for controlling UAVs is shown in Doherty et al. (2015), where a distributed architecture designed to provide a set of necessary functions for missions requiring a high level of independence is discussed, such as, for example, in a rescue situation where the drones identify injured people or in a more traditional problem of delivering packages. The architecture in question is classified as HDRC3 (Hybrid Deliberative/Reactive) with a structure of 3 conceptual layers: the control layer, orientated at managing the UAVs, the reactive layer, for high level coordination by means of behaviors such as fly-to or scan-area and, finally, the deliberative layer, which provides the highest level algorithm, such as, for example, planning tasks.

Some research on a multi-agent architecture level is focused on solving more general problems, such as, for example, avoiding collisions (Sislak et al., 2007), planning routes (Moon et al., 2013) or identifying specific objectives (Dasgupta, 2008). This last piece of research stands out for setting a multi-agent architecture based on swarm-computing in order to automatically detect objectives by means of UAVs in a decentralized fashion.

Table 1 shows the advantages and disadvantages of the most relevant works cited in this article.

3. Proposed architecture

3.1. Motivation

Within the context of analyzing and monitoring areas affected by catastrophes, most of the research there is in the literature, linked to designing architectures which support the display of multi-robot systems orientated towards analyzing events, usually focuses on specific functions or solving specific problems, such as those mentioned in the previous section. This article provides a basis on which to display systems made up of mobile robots which cooperate in order to monitor environments, but greater generality and scalability are required from the perspective of analyzing events of interest.

Our contribution, to be precise, focuses on designing a new multi-agent architecture which, from a general perspective, covers the problem of information analysis and crisis management in these types of scenario. Therefore, a series of agents and communication mechanisms have been specified which are general enough to cover the monitoring of catastrophe-prone environments. These agents intervene in the process of physically controlling the UAVs, when processing the information obtained by the sensors of these UAVs, in the intelligent analysis and information fusion in order to understand what is happening around them, and, finally, in supporting the decision-making process for the human workers.

We also intend the architecture put forward herein to be of an integrative nature, so it is relatively simple to incorporate both new physical monitoring devices and new algorithms which are capable of analyzing new situations or environments. We have based this objective on a clear definition of the roles that different architecture components have and on a formal normality analysis model (Albusac et al., 2009) which is already in use in the field of intelligent surveillance of urban traffic environments.

As regards systematic requirements, our architecture has been designed to have those listed below:

- Availability, linked to the robustness of the system displayed from the architecture, fault detection and consideration of the consequences these could have. By way of example, the architecture should contemplate what happens when a UAV breaks down.
- Evolvability, understood as the capacity the architecture has to respond in the face of changes to hardware or software. One possible example would be modification of the internal behavior of any of the architecture agents.
- Integration, linked to the capacity the architecture has of integrating new devices. One example in this context would be using more modern UAVs.
- Manageability, referred to the interaction between human staff and the system displayed from the architecture. This requirement is related to the use of graphical monitoring tools.
- Scalability, conceived as the capacity and the mechanisms devised in the architecture for integrating new components. One of the aspects best representing this requirement would be the possibility the architecture has of providing support to new types of intelligent analysis.
- Security, linked to the mechanisms provided for preventing inappropriate or unauthorized use of the system.

How these systematic requirements are fulfilled will be reflected in Sections 3.2 and 3.4, in which the design of the multi-agent architecture is addressed, and in Section 4, in which a specific case study is discussed in detail.

3.2. Architecture overview

The multi-agent architecture set out in this paper is shown graphically in Fig. 1. This architecture enables multi-agent systems to be deployed in order to analyze the information obtained in environments affected by catastrophes and to provide support in the decision-making process. This architecture is structured in three well defined layers: the *information retrieval layer*, the *cognitive layer* and the *user layer*. The components of the different layers are communicated through a communications middleware that abstracts the architecture from the programming languages, operating systems and communication networks involved when deploying the architecture. Below, the main features of each layer are introduced.

The *information retrieval layer* is that responsible for obtaining information about the physical environment intended to be analyzed. The mobile robots obtain this information by means of the sensors integrated into them. Each proxy agent controls a mobile robot, thereby recreating an abstraction layer which enables the hardware part (mobile robot) to be independent from the software part (proxy agent). Although in this paper the mobile robot is represented by a drone, the abstraction the mobile robot establishes enables terrestrial robots to be used.

The *cognitive layer* is the most complex layer of the architecture and its aim is to carry out an intelligent analysis of what is taking place in the environment which is being monitored. Therefore, the intelligent agents of this layer use the information created by the information retrieval layer to interpret situations and events, both from a local and global perspective. The final output from this layer consists in a set of recommendations or actions the human operator may follow to deal with the catastrophe that has occurred.

The *user layer* is that which potentially integrates monitoring tools used by human operators, which receive the information created by the remaining layers. This information may be direct, such as that provided by a drone camera or obtained as a result of intelligent analysis, such as, for example in estimating the risk when a smoke cloud spreads. Also, recommendations of actions to take given by the artificial system to the human staff are considered in this layer.

The architecture considers the integration of agents responsible for managing the multi-agent system, which provide essential services such as a white pages service, a yellow pages service, and a direct communication service. These agents comply with the recommendations made by FIPA to promote agent technology and interoperability.

Moreover, and in order to make it easier for the reader to understand how the information is transformed as a result of the intervention of different agents in the architecture, the Fig. 2 shows how this flows. The different high-level stages are numbered in this figure: (1) information retrieval, (2) hardware abstraction and notification, (3)

Table 1
Table that summarized the advantages and disadvantages of the cited works (UAVs, disaster management and multi-agents).

UAVs and disaster management			
Ref.	Authors	Advantages	Disadvantages
(Xu et al., 2014)	Xu et al.	Adjustment for imaging resolution; photogrammetric workflow applied; experiments with flying height	Focused on one UAV; scalability; no integration with entities that analyze different events of interest.
(Baiocchi et al., 2013)	Baiocchi et al.	Stereoscopic reconstruction; flight planning optimization;	No automatic control; single UAV; scalability; no integration with entities that analyze different events of interest.
(Chou et al., 2010)	Chou et al.	Image rectification integration; estimation of collapsed lands	Single drone; scalability; no integration with entities that analyze different events of interest.
(Torres et al., 2016)	Torres et al.	Coverage path planning integration; 3D reconstruction of environments; battery optimization	Single drone; scalability; only focused on coverage.
(Fikar et al., 2016)	Fikar et al.	Promote coordination between rescue teams; simulations with a large number of UAVs/agents	Dependent on initial vehicles/UAVs location; system does not learn; focused on goods management .
(Zhao et al., 2019)	Zhao et al.	Similar performance to centralized approaches; addresses distributed information fusion and decision-making	Focused on tracking ground vehicles; no integration with entities that analyze different events of interest.
(Andriluka et al., 2010)	Andriluka et al.	Automatic victim detection; combination of multiple detectors	Indoor experiments not realistic; single drone; scalability; no integration with entities that analyze different events of interest.
(Portmann et al., 2014)	Portmann et al.	New dataset on thermal image sequences; methodology for image analysis	Focused on tracking people; no testing with UAV cameras.
(Kurnaz et al., 2010)	Kurnaz et al.	Fuzzy control to deal with uncertainty and vagueness; simple yet efficient approach	Focused on UAV control; need to address unstable flying conditions.
(Sun et al., 2011)	Sun et al.	Addresses physical properties of real environments; pre-processing stage to deal with complexity; can tune factors depending on the task requirements	Focused on UAV flight routes; single drone.
(Ji et al., 2015)	Ji et al.	Cooperative search based on asynchronous, distributed approach; handles nonconvex environment with arbitrary obstacles; communication issues are addressed	No tests with real drones or UAVs-related simulation software; dynamic conditions are not dealt with.
UAVs and multi-agents			
Ref.	Authors	Advantages	Disadvantages
(Arokiasami et al., 2016)	Arokiasami et al.	Generalization approach for obstacle detection and avoidance; behavior-based approach to promote scalability	JADE (Java)-dependent; no source code is provided to test the framework; no methodology discussed to include new agents.
(Doherty et al., 2015)	Doherty et al.	Scalability to deal with multiple events of interest; specific tool to handle high-level tasks; robust integration and interoperation between modules	No tests/simulations with multiple UAVs; data specification and knowledge flow could be generalized.
(Sislak et al., 2007)	Sislak et al.	Experiments run on ATC framework; integration of a rule-based solver algorithm; deals with both cooperative and non-cooperative collision avoidance	Focused on collision avoidance; no integration with entities that analyze different events of interest.
(Moon et al., 2013)	Moon et al.	Handles multiple UAVs; computational costs are addressed; tests run within the context of cooperative missions in dynamic environments	Focused on task assignment and path planning; no integration with entities that analyze different events of interest.
(Dasgupta, 2008)	P. Dasgupta	Distributed system for automatic target recognition; resource constraints are addressed; scalability regarding the number of UAVs	Focused on object tracking; no integration with entities that analyze different events of interest.

event detection and analysis, (4) information fusion, and (5) decision support.

Initially, the mobile robots obtain information about the area around them by means of their sensors, such as, for example, a video camera (see step 1 in Fig. 2). The proxy agents, apart from controlling the mobile robots, encapsulate this information in events which are sent via events channels to the information analysis agents (see step 2 in Fig. 2). These agents are already capable of analyzing the information in order to infer what is taking place around them and use a knowledge base for this purpose. In this research work we suggest using a formal model of normality analysis (Albusac et al., 2009) as a tool for defining the previously mentioned knowledge bases (see step 3 in Fig. 2).

The partial visions of the world are reflected in the blackboard architecture, in such a way that the translator agents and the fusion agents can create a global view of the area under analysis, if this is necessary. In this respect, the fusion management agent is that responsible for deciding if information fusion is required, according to the knowledge put on the blackboard by the previous agents (see step 4 in Fig. 2). Finally, the decision support agent uses this knowledge to send out alerts and suggest possible decisions to the human staff by means of the user layer (see step 5 in Fig. 2).

All these architecture components, including the communication mechanisms used, are formalized and discussed in the following subsections.

3.3. Architecture description and formalization

Next, we introduce the formalization of proposed architecture to deploy multi-agent systems in environments affected by catastrophes. This can be defined as the tuple

$$A = \langle E, C, EC, IRL, CL, UL \rangle$$

where

- E is the environment where a catastrophe might take place, which can be considered as a set of points of interest that must be analyzed or monitored, $E = \{p_1, p_2, \ldots, p_x\}$. This definition of environment, according to a set of points of interest, is carried out in an initial stage where the environment is characterized.

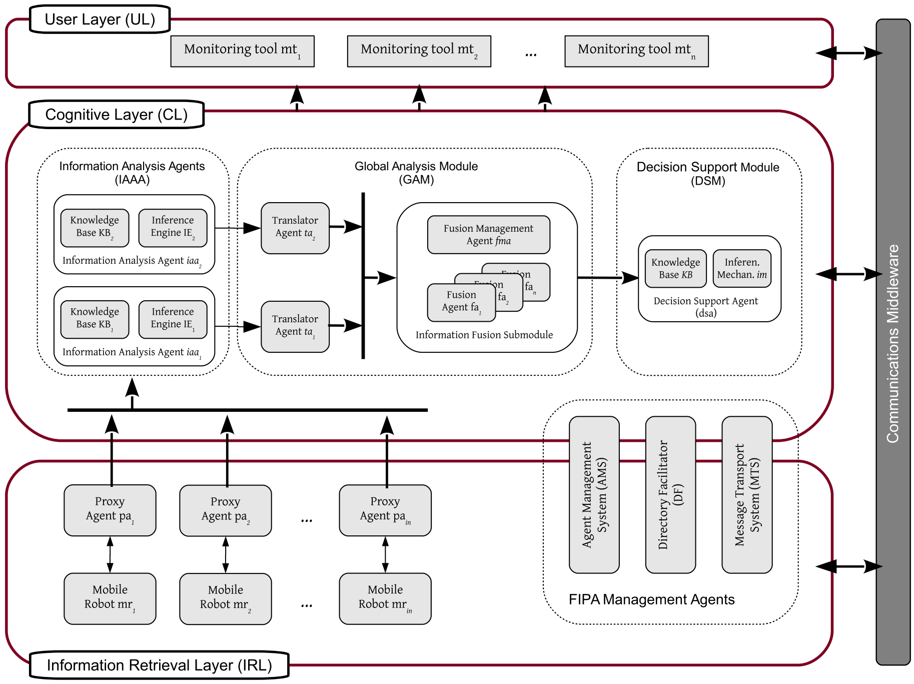

Fig. 1. Global overview of the multi-agent architecture proposed for monitoring environments affected by catastrophes.

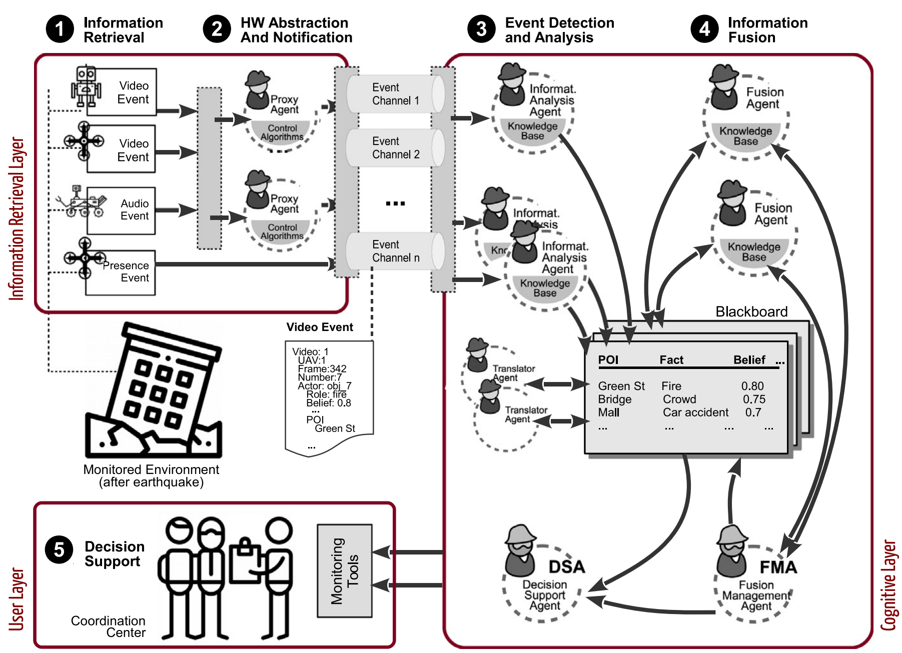

Fig. 2. Global overview of the information flow between the major components of the proposed multi-agent architecture.

- C is the set of concepts or events of interest to be monitored, so that $C = \{c_1, c_2, \dots, c_m\}$. Each c_i is defined as $c_i = \langle V_i, DDV_i, KB_i, ie_i \rangle$ where V_i represents the set of variables used to define the concept c_i (i.e. $V_i = \{v_{i1}, v_{i2} \dots, v_{in}\}$) and DDV_i is the definition domain of the variables that are involved in V_i, (i.e. $DDV_i = \{DDV_{i1}, DDV_{i2}, \dots, DDV_{in}\}$). This way, DDV_{ij} represents the values that the variable v_{ij} can take when analyzing the concept c_i. Finally, KB_i is the knowledge base that allows to perform the analysis of the concept c_i, while ie_i is the inference engine that allows to determine what is happening according the concept c_i. In Section 5, the design of a concept devised to monitor the propagation of toxic smoke in an environment affected by a catastrophe is carried out.
- EC is the set of event channels deployed so that the architecture agents can publish or subscribe to information over a certain concept of interest c_i (i.e. $c_i \in C$), $EC = \{ec_1, ec_2, \dots, ec_y\}$. Event channels allow the deployment of a many-to-many communication mechanism where there is an independence between the agents that send information and those that receive it.
- IRL is the information retrieval layer of the architecture.
- CL is the cognitive layer of the architecture.
- UL is the user layer of the architecture.

Next, we go in depth in the formalization of the layers that compose the architecture.

The Information Retrieval Layer (IRL) is defined as the tuple

$$IRL = \langle MR, PA, \mathscr{P}(MR \times PA \times EC) \rangle$$

where

- MR is the set of mobile robots responsible for examining the points of interest of the environment to collect data, so that $MR = \{mr_1, mr_2, \dots, mr_n\}$. This approach allows to make decisions in the upper levels. The mobile robots can be aerial or ground, and integrate a set of sensors of different kinds to gather data from the environment, depending on the kind of variable to be measured. An example of mobile robot might be an aerial drone that integrates a camera to record images.
- PA is the set of proxy agents responsible for establishing how the mobile robots move and operate on the environment, making an abstraction layer over mobile robots, $PA = \{pa_1, pa_2, \dots, pa_m\}$. Each proxy agent will provide the mechanisms required to control a specific kind of mobile robot.
- $\mathscr{P}(MR \times PA \times EC)$ will be composed of trios (mr, pa, ec) where the first item mr is a mobile robot (i.e $mr \in MR$), the second item pa is a proxy agent that controls the mobile robot (i.e. $pa \in PA$) and the third item ec is the event channel used by the proxy agent to publish information.

The Cognitive Layer (CL) is defined as the tuple

$$CL = \langle IAA, B, GAM, DSM \rangle$$

where

- IAA is the set of information analysis agents that will analyze the environment, $IAA = \{iaa_1, iaa_2, \dots, iaa_z\}$. Each agent iaa_i is a trio (c, e, ec) that will analyze a certain concept c (i.e. $c \in C$) regarding a specific point in the environment (i.e. $e \in E$), using an event channel ec to collect the data to be analyzed (i.e. $ec \in EC$).
- B represents the blackboard architecture used to store the data and hypothesis (partial knowledge) obtained by the agents IAA regarding the studied concepts C in the environment E.
- GAM is the global analysis module that will obtain general conclusions from the partial knowledge generated by the IAA, making use of the blackboard architecture. It is defined as $GAM =< TA, fma, FA >$, where
  - TA is the set of translator agents that will obtain global representations of information for each concept c_i of C, from the information available on the architecture regarding such concept. Thus, $TA = \{ta_{c_1}, ta_{c_2}, \dots, ta_{c_m}\}$, being ta_{c_i} the translator agent responsible for obtaining the global representation of information for the concept c_i.
  - *fma* is the fusion management agent, responsible for deciding whether it is required or not to aggregate the knowledge associated to a concept c_i to infer global conclusions.
  - FA are the information fusion agents in charge of making the information fusion, if this is required. $FA = \{fa_{c_1}, fa_{c_2}, \dots, fa_{c_m}\}$, where fa_{c_i} is the information fusion agent for the concept c_i.
- DSM is the decision support module, whose goal consists in giving warnings and making recommendations (interventions or actions) to the human personnel responsible for managing crisis situations caused by the catastrophe. It is composed of a single agent named decision support agent, noted as *dsa* and defined by the tuple: $dsa = \langle KB, im \rangle$, where KB is the knowledge base of the agents and im is the inference mechanism that allows to generate the recommendation.

Finally, the User Layer (UL) is composed of a set of monitoring tools $UL = \{mt_1, mt_2, \dots, mt_x\}$ that will be used to visually render the information generated by the different agents that are integrated in the architecture.

3.4. Communication

In the multi-agent architecture under consideration 3 different types of communication, thought up for providing flexibility when sharing information and for supporting cooperation among agents are proposed. Below, the main features of each communication layout are described:

- Direct communication enables an agent to send a message to another one by using its physical address in the message itself. In other words, in this type of communication intermediaries are not needed. In the setting of the architecture put forward herein, this layout could be appropriate when there is no cooperation among various agents, or if high level information is dispatched, such as what occurs between the fma and the dsa. In short, this mechanism provide one-to-one communication.
- Communication based on events channel, which enables communication guided by the contents of the messages and not by their emitters or recipients. This type of communication promotes architecture scalability, since it is possible to define roles based on publication and subscription to messages by the main events channel. Therefore, for example, a iaa_i could subscribe to an events channel ec_j in which a subset of proxy agents $PA_k \subseteq PA$ publish a certain type of information, linked to a concept $c_i \in C$. In other words, this communication mechanism represents many-to-many communication, providing independence between publishers and subscribers and offering a communication driven by the message content.
- Communication by means of a distributed blackboard architecture, which enables a set of agents to share knowledge from information compiled about the area affected by the catastrophe. Typically, the blackboard will be used so that the set of information analysis agents IAA write the different perceptions of the monitored scene. In other words, the blackboard acts as a repository of knowledge that the agents incrementally use to understand what takes place in the monitored environment. Thanks to this communication layout, the information fusion agents FA can provide high level knowledge about what is happening around them, from a global perspective.

Since the components of the architecture must interact with one another considering the previously mentioned communication mechanisms, in this work a communication middleware is proposed to connect the different layers (see Fig. 1). From a practical point of view, ZeroC ICE is recommended (Henning, 2004), which represents a modern object-oriented server–client RPC framework with a number of services that directly support some of these communication types, such as the one based on event channels.

4. Case study: deploying a multi-agent system to monitor an industrial complex and a urban environment

In this section, a case study set in a city which is relatively close to an industrial complex, is discussed. Fig. 3 shows an aerial view of this scenario, where in the upper part the distance there is between both of these can be seen and in the lower part a hypothetical situation of fires which have broken out at the petrochemical plant and at the refinery in the industrial complex has been recreated. These fires create toxic smoke clouds which could potentially reach the city.

Therefore, in this section, the steps associated with the deployment, from the proposed architecture, of a system which is capable of obtaining information and assessing the area affected by this catastrophe in the context of spreading toxic smoke are defined. Additionally, monitoring the sports complex situated in the city has been contemplated, as in times of crisis people may potentially crowd together there.

In the scenario recreated in Fig. 3 it is essential to master two fundamental matters: (i) managing the evacuations on the complex premises (ii) alerting the population in the nearby city. In both matters

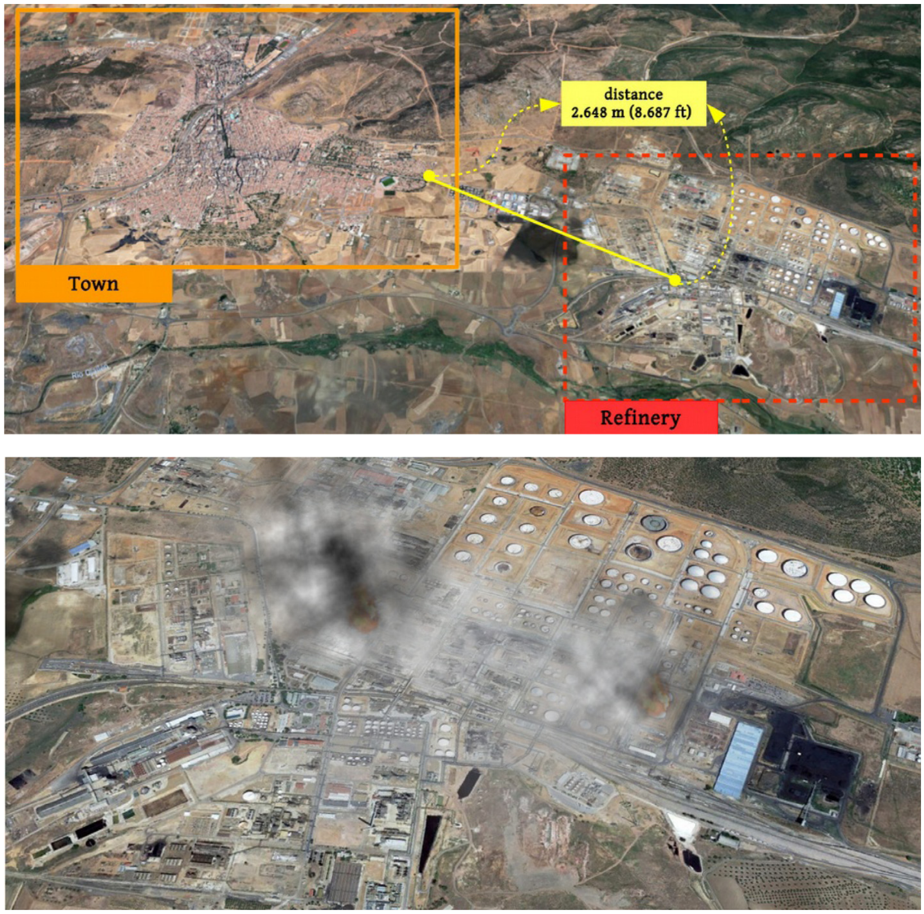

Fig. 3. Above: aerial view of the city and industrial complex, separated by approximately 2.6 kms. Below:aerial view of an industrial complex. Recreation of the fires which have broken out in the petrochemical plant (left) and at the refinery (right), sources of the toxic smoke cloud.

time is essential, so a system that is capable of analyzing information about the environment quickly and practically and using it to support a crisis management plan would be of great help.

Below, the general steps are introduced which are considered for deploying the monitoring system, among which the following are mainly noteworthy:

1. Characterization of environment E by means of defining the exit roads in the 3D space which may be especially important when it comes to obtaining and analyzing information. Identification of areas that are especially critical, such as, for example a place with a high risk of fire or a space in which many people crowd together.
2. Definition and instantiation of the normality components which integrate the concepts specifications or events of interest C used to analyze the environment E by the information analysis agents IAA. Some typical examples could be the analysis component from the spread of toxic smoke or from crowds of people, situations which commonly occur when a catastrophe strikes. This specification of concepts of interest is associated with the definition of the event channels EC, used by the agents to publish or receive information about these concepts.
3. Specification and deployment of mobiles robots and processing agents integrated into the information retrieval layer (IRL). In this step creating the event channels EC is contemplated as well as associating the processing agents PA with these.
4. Specification and deployment of agents of the cognitive layer (CL). In this step, creating the blackboard architecture B is contemplated as well as subscription of the information analysis agents IAA to the event channels EC.
5. Specification of the high-level monitoring mechanisms integrated into the user layer (UL).

Below, details of these steps will be given within the context of a case study set in this section. On this matter, a non-specialist person responsible for deploying a system from the proposed architecture could rely on predefined templates that the middleware ZeroC ICE provides for activating processing nodes, servers and services. The latter wrap the functionality provided by the designed agents. Scripts can also be given to automatically deploy the system when required, while one default configuration regarding the agents that analyze the environment can be established. However, this person must define the environment to be monitored through the points of interest, $E = \{p_1, p_2, \dots, p_x\}$, using and XML editor. Surveillance components must be designed and developed before deploying the system. Thanks to the use of fuzzy logic, these are readable by non-experts and can be easily adjusted to improve, if need be, the representation of normal/abnormal situations.

4.1. Environment characterization

As has already been seen in Section 3.3, the environment to be analyzed has been defined as $E = \{p_1, p_2, \dots, p_n\}$, in such a way that each p_i represents a specific exit road within the area. The exit roads will to a large extent depend on their physical features and the type of catastrophe which a priori could occur in this environment. Due to the information retrieval from the environment which would essentially be carried out by drones (and these are usually planned based on routes defined by a series of points in a 3D space) it seems appropriate for the basic characterization of the environment to be done like this.

In the context of the case study set, Fig. 4 graphically reflects 6 exit roads: p_1 and p_2 are associated with monitoring points linked to the petrochemical plant and the refinery respectively, while p_3, p_4, p_5 and p_6 are linked to the exit routes r1, r2 and r3, used to evacuate the sports complex located in the area closest to the city. To monitor the first 5 points a mini-fleet of 3 drones has been considered (d_1, d_2 and d_3), in such a way that d_1 and d_2 are responsible for monitoring PoI_1 and PoI_2, whilst d_3 would be allocated a mission with points PoI_3, PoI_4 and PoI_5. Points of interest are defined in advance, previously to the occurrence of an emergency. These points are actually the waypoints that compose the mission that are loaded into the UAV and controlled by the associated proxy agent. In the current version of the software prototype, points of interest, and their characterization, are specified in XML files.

Furthermore, monitoring a sports complex located in the city has been contemplated by means of another mini-fleet of 3 drones (d_4, d_5 and d_6). This sports complex can be seen in Fig. 5.a. This complex is made up of 3 football fields, with seating for 4500 people, 3 open-air sports tracks, 1 indoor sports hall which can hold up to 500 people, 1 indoor swimming pool and a range of tennis and paddle tennis courts. In this sub-scenario, the consequences the potential spread of the toxic cloud from the industrial complex has can be hypothetically contemplated. As the sports complex may be full of people, it is essential to obtain useful information for emergencies. In this event, and instead of carrying out an exhaustive study of what is taking place in the environment based on the definition of the normality components which encapsulate concepts or events of interest, the goal is to monitor the environment considering all the important exit roads.

As shown in Fig. 5.c, the exit roads in the defined environment have been allocated a numerical priority, which varies between 1 and 3. The priority 1 p_i are set at the ends of the sports complex, in such a way that in the event of emergency the drones first go to the most distant locations. In a second and coordinated pass, the drones would start closing their radius of action, linking the priority 2 p_j. Finally, the priority 3 p_k reflect a much higher closeness level to the premises than the others.

4.2. Definition of normality components

The definition of normality components implies specifying or learning the necessary knowledge for the information analysis agents to be capable of understanding what is happening in the environment. In this case study, a formal model has been chosen (Albusac et al., 2009) which has enabled the c_1 concept to be specified. This concept is linked to the normality analysis in relation to the spread of toxic smoke, discussed further on in Section 5.

This specification is based on the definition of a set of variables V_i, variable definition domains DDV_i, knowledge base KB_i, and inference engine ie_i, being all those elements included in the component specified. It is important to stress that the general definition of the component can be reused in a range of environments. In other words, it is possible to reuse the set of variables V_i, the knowledge base KB_i and the inference engine ie_i. However, when instancing the component within a specific scenario, it will be necessary to specify the range of valid values, represented by the variable definition domains DDV_i.

In Section 5, the component responsible for monitoring the spread of toxic smoke is not just defined, but is also instanced, that is, its details will be given, with specific values, for the scenario under analysis.

4.3. Information retrieval layer

As regards obtaining the information about the environment, the proxy agents PA are of note, which are responsible for the mobile robots MR. Essentially, a proxy agent pa_i is capable of, from a software point of view, controlling a subset of mobile robots MR_i, which are typically drones (a proxy agent could be a Unix process composed of multiple threads, one per mobile robot). This approach enables a software abstraction layer to be established over the different specific drone or robot models which can be used when retrieving information about the environment by sensors, such as, for example, with video cameras. In other words, if the model of a drone or a determined sensor changes, then all that is needed will be to implement a new proxy agent pa_i which can adequately control them.

Algorithm 1 Pseudo-code of the pa_i that controls a single mr_j

Require: m_i `// initial mission allocated to the proxy agent`
Require: mr_j `// mobile robot controlled by the proxy agent`
Require: MAX_TIME_POI `// max waiting time in PoI`
Require: EC `// list of event channels to publish information`

```
for wp = m_i.waypoints(1) to m_i.waypoints(n) do
  MoveMobileRobot(mr_j, wp)
  if Arrived(mr_j, wp) then
    time_in_POI ← 0
    while time_in_POI < MAX_TIME_POI do
      image ← CaptureImageFromCamera()
      for ec = EC(1) to EC(n) do
        SendCapturedImage(image, ec)
      end for
      time_in_POI ← time_in_POI + IncreaseWaitingTime()
      Sleep(GetWaitingTime())
    end while
  end if
end for
```

In this case study, deploying 2 proxy agents will be considered: pa_1 and pa_2. The former will be responsible for controlling the 3 drones d_1, d_2 and d_3, deployed in the industrial complex, while the latter will control the other 3 drones d_4, d_5 and d_6, connected to the sports complex in the city. This decision has been justified on two grounds. The first is the physical distance there is between the industrial complex and the city makes it necessary for each proxy agent to be deployed in a processing node close to the physical drones. The second lies in the practicality of maintaining two proxy agents which publish information from different typology in the events channel which connects the information retrieval layer with the cognitive layer.

In terms of operating, a proxy agent pa_i resolves the problem of guiding the drones by considering a list of exit roads in the 3D space, known as waypoints. On a basic coordination level, each pa_i is initially allocated a list of waypoints ordered by priority, called missions, in such a way that the set of drones destined for monitoring a scenario will cover all its exit roads. If just one physical drone were available to carry out an analysis of the environment, then this would be responsible for all the exit roads in this environment. The missions may change dynamically according to the monitoring needs of the environment. Additionally, it is possible to define missions which contain a sole exit road so that this can be continuously monitored. Algorithm 1 shows the pseudo-code of the proxy agent pa_i that controls a single robot mr_j.

Proxy agents can be replicated using a master–slave approach based on the bully algorithm (Garcia-Molina, 1982). Thus, a scenario where multiple replicas of one proxy agent actually control one drone, but acting as one and increasing the system robustness in case an unexpected error takes place. This approach is proposed instead of considering a situation where one drone cannot be controlled by a proxy agent and a different drone must cover the points of interest assigned to the drone whose proxy agent failed.

Finally, in terms of communication, deploying two events channels ec_1 and ec_2 is considered. ec_1 will be used for the proxy agent pa_1 for

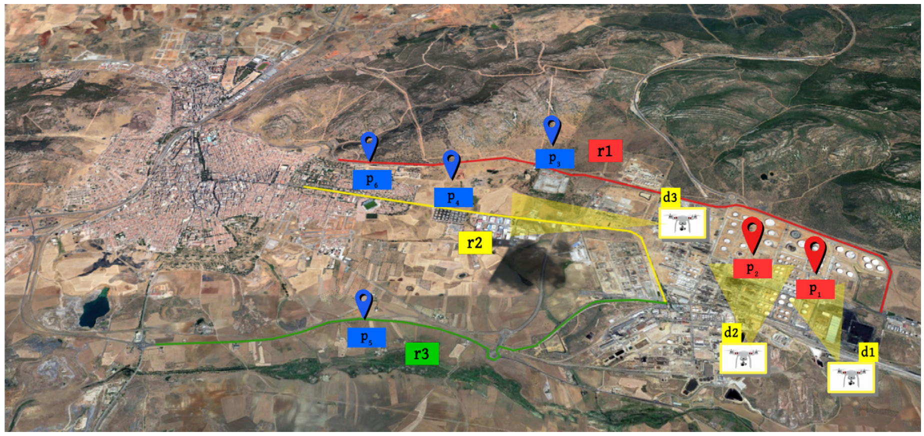

Fig. 4. Overview of the main roads (r1, r2 and r3) connecting the industrial complex with the city. In red the exit roads are shown from the perspective of monitoring critical elements: p_1 is associated with the petrochemical plant and $p2_2$ with the refinery. In blue the exit roads are shown from the perspective of monitoring the evacuation roads and the city: p_3, p_4, p_5 and p_6. Three drones $d1$, $d2$ and $d3$ have been considered for analyzing points p_1 to p_5.

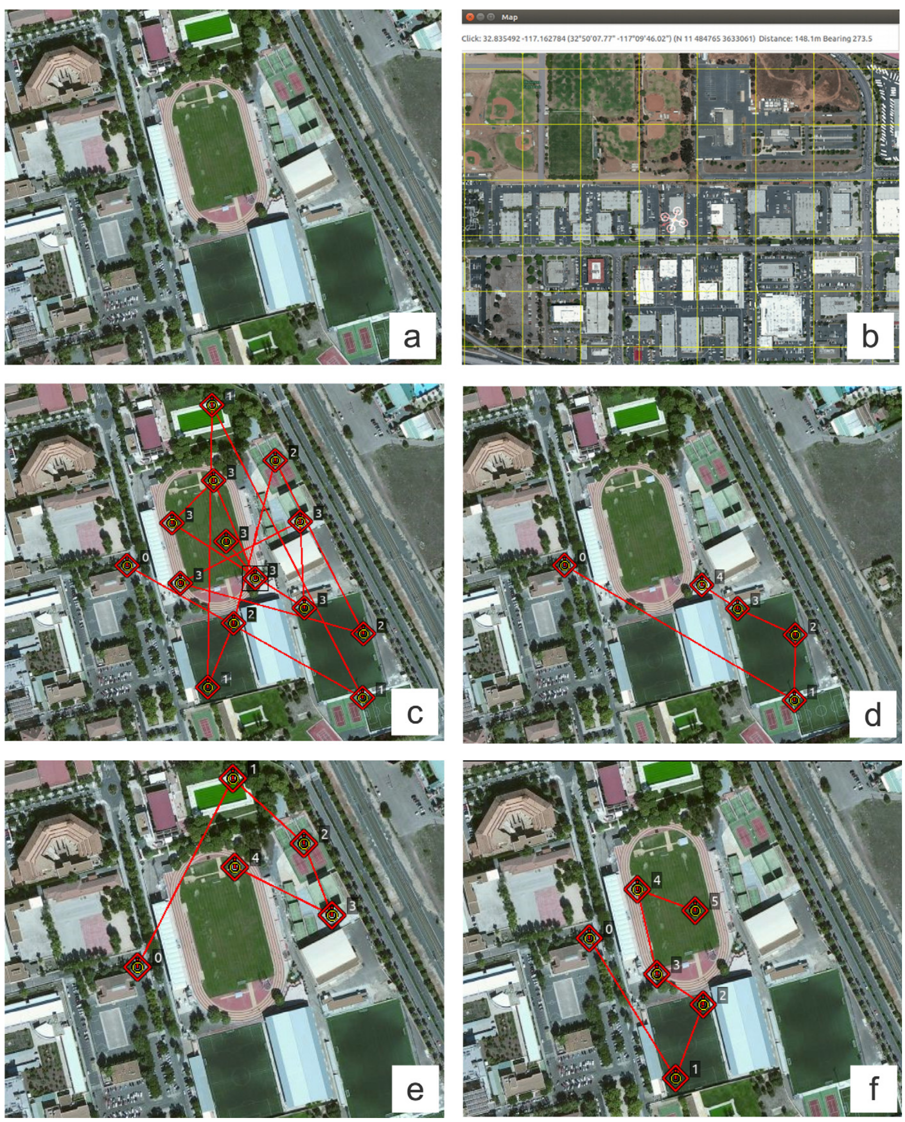

Fig. 5. Different screen shots linked to the monitored scenario. (a) Bird's eye view of the sports centers linked to the case study. (b) Graphical look of *Software in the Loop*, the simulation tool used in the case study. (c) Design of the mission based on the different exit roads. (d) Mission assigned to drone 1. (e) Mission assigned to drone 2. (f) Mission assigned to drone 3.

Algorithm 2 Pseudo-code of the information analysis agent iaa_l

Require: c_i `// concept to be analyzed by the agent`
Require: e_j `// specific environment point`
Require: ec_k `// event channel to collect information`
Require: B `// Blackboard to publish information`
while $TRUE$ **do**
  $event \leftarrow ReceiveEventFromEventChannel(ec_k)$
  $degree\_of\_normality \leftarrow AnalyzeEvent(event, c_i, e_j)$
  $degree\_of\_risk \leftarrow ComputeRisk(degree\_of\_normality)$
  $WriteOnBlackboard(B, c_i, e_j, degree\_of\_risk)$
end while

publishing information relating to the images which will subsequently be analyzed in order to detect the potential spread of fire and toxic smoke. ec_2 will be used by the proxy agent pa_2 in order to publish information linked to the images obtained of the sports complex. Although the type of event in both situations is associated with images captured by drones, we can make out two different events channels in order to delimit the extent to which the information spreads according to the monitoring needs.

4.4. Cognitive layer

As regards intelligent processing of the data previously captured on the information retrieval layer IRL, and supplied by the proxy agents PA, the deployment of the information analysis agents IAA is of note. Essentially, for each exit road p_i which the environment characterization is made up of (see Section 4.1) it will be necessary to deploy one or various information analysis agents, depending on which normality components are going to be analyzed. If both the spread of fire and the toxic smoke are analyzed, then two information analysis agents will have to be deployed for each exit road p_i.

In this case study the 6 exit roads contemplated in the Fig. 4 have been considered. p_1 and p_2 are associated with the petrochemical plant and the refinery respectively. For these two exits, the analysis of the spread of toxic smoke has been considered, and therefore 2 information analysis agents iaa_1 and iaa_2 are necessary which instance the concept c_1. In this paper, it is assumed that if the smoke reaches a critical point, then it may be thought that the fire itself may reach this point in the near future. Moreover, p_3, p_4 and p_5 are associated with the connection routes to the city. For these three roads the analysis of the spread of toxic smoke has been considered, so the c_1 concept must be instanced once per exit road, thereby creating the necessary knowledge for the information analysis agents iaa_3, iaa_4 and iaa_5. All these agents share one sole blackboard B (Jennings et al., 1998) and are subscribed to the events channels ec_1,in which the proxy agent pa_1 will publish the information associated with the visual events created from the data obtained by the drones d_1, d_2 and d_3. Algorithm 2 shows the pseudo-code of one information analysis agent that monitors the normality of a concept c_i at a specific location e_j, while Algorithm 3 delves into the calculation of the degree of normality regarding the concept c_i (toxic smoke propagation), which is defined in Section 5 according to the adopted normality model (Albusac et al., 2009).

Moreover, p_6 is associated with the sports complex, but also represents a nearby area in respect to the industrial complex. In this case study it has been assumed that 3 information analysis agents, iaa_0, iaa_1 and iaa_2 are deployed, which are linked to the sports complex, one for each drone in the second mini-fleet (d_4, d_5 and d_6). In principle these drones will be associated with instancing a concept which will enable crowds of people at specific physical points to be analyzed. However, exclusively dedicating an agent to analyzing the potential spread of toxic smoke could be considered, in an identical way to that previously discussed with information analysis agents.

Algorithm 3 Calculation of the degree of normality $N^l_{c_i}$ by an iaa_l

Require: Φ_{C_i} `// Set of normality restrictions of` c_i
Require: *distance* value v_1
Require: *speed* value v_2
Require: *direction* value v_3
Require: *spreading* value v_4
// Fuzzification of variables
$v'_1 \leftarrow fuzzification(v_1, DDV_1)$
$v'_2 \leftarrow fuzzification(v_2, DDV_2)$
$v'_3 \leftarrow fuzzification(v_3, DDV_3)$
$v'_4 \leftarrow fuzzification(v_4, DDV_4)$
// being v'_1, v'_2, v'_3, v'_4 fuzzy variables composed by set of tuples whose form is (*fuzzy set, membership value*).
for each rule r_j defined in Φ_{c_i} **do**
  $\mu_j \leftarrow FuzzyLogicController(v'_1, v'_2, v'_3, v'_4, r_j)$
  // being μ_j the degree of satisfaction of rule r_j
end for
$N^l_{c_i} \leftarrow defuzzification(\mu_1, \mu_2, ..., \mu_n)$ //degree of normality associated to a concept c_i for an information analysis agent iaa_l.

The simulations carried out with the software SITL[1] (*Software in the Loop*), in which the analysis times of the images captured by the 3 drones, respectively, have been disregarded, have in total clocked up a time for going to all exit routes in the mission of 4 min and 30 s. In other words, considering deploying the drones automatically, it is possible to retrieve information about the environment affected by a hypothetical natural catastrophic in hardly 5 min. This information may be highly useful for emergency teams arriving on the scene.

As for fusing information, linked to the GAM global analysis model, in this case study only deploying the translator agent ta_i is contemplated, which is responsible for obtaining a homogeneous representation of the information created by the IAA with respect to a global space of coordinates, and a fusion management agent fma. By way of example, the hypothetical situation in which the 2 drones d_1 and d_3 monitor the spread of a cloud of toxic smoke which moves towards the city from p_4 and p_5, respectively, has been contemplated. The level of risk calculated by the information analysis agents, which in the last instance are responsible for the analysis, can be added to obtain more accurate knowledge about what is happening.

Thus, the degree of normality obtained by each iaa_j can be used to compute the degree of abnormality as follows:

$$A^j_{C_i} = 1 - N_{c_i}$$

where A represents the degree of abnormality, j the index of the information analysis agent that carries out the calculation, and c_i the normality component related to the analysis.

On the other hand, each iaa_j has associated a weight w_j that determines the importance of its analysis within the context of the global normality analysis. Such weight varies dynamically, so that those information analysis agents that detect an anomaly will hold a higher w_j when assessing the global normality value. The variation of this weight depends on the variation of the degree of abnormality regarding the normality definition.

$$\triangle(A - N)^j_{c_i} = A^j_{c_i} - N^j_{c_i}$$

A negative difference implies that the degree of normality is greater than the degree of abnormality and, therefore, it is not necessary to increment the weight. On the contrary, a positive increment means that the anomaly grows above the normality, and the weight w_j associated to the iaa_j should have a higher relevance when computing the global

[1] http://ardupilot.org/dev/docs/sitl-simulator-software-in-the-loop.html.

Algorithm 4 Global calculation of the degree of abnormality

Require: $N^j_{C_i}$ `// Every degree of normality` N `associated to each` iaa_j `and a Normality Component` c_i
for all $N^j_{c_i}$ **do**
$A^j_{c_i} \leftarrow 1 - N_{c_i}$ // Degree of abnormality calculated by an iaa_j
$\triangle(A-N)^j_{C_i} \leftarrow A^j_{C_i} - N^j_{C_i}$ // Variation between the degree of abnormality and the degree of normality.
$w'^j \leftarrow w^j + f(\triangle(A-N)^j_{C_i}, w^j)$ // Dynamic weight adjustment
end for
$w'^j \leftarrow w'^j / \sum_{j=1}^{n} w'^j$ // Normalization of weights
$global\_degree\_of\_abnormality \leftarrow \sum_{i=1}^{n} A^j_{C_i} \times w'^j$

assessment. The function that is used to compute the weight increment is as follows:

$$f(\triangle(A-N)^j_{c_i}, w^j) = \begin{cases} 0 & \text{if } (\triangle(A-N)^j_{c_i}, w^j) \le 0 \\ (A-N)^j_{c_i} & \text{Otherwise} \end{cases}$$

Therefore, the variation w'^j of a weight w^j associated to an iaa_j is computed as follows:

$$w'^j = w^j + f(\triangle(A-N)^j_{C_i}, w^j)$$

Once the new weights have been computed depending on each iaa_j regarding the variation of anomalies, a normalization process is carried out so that the global sum of all the weights is equal to 1.0.

$$w'^j = w'^j / \sum_{j=1}^{n} w'^j$$

being n the total number of information analysis agents.

Finally, the global degree of abnormality is equal to the weighted average of the degrees of abnormality computed by each iaa_j multiplied by the computed weights.

$$\sum_{i=1}^{n} A^j_{C_i} \times w'^j$$

After discussing these concepts, the information fusion to obtain the global degree of abnormality is performed as follows:

4.5. User layer

The high-level monitoring mechanisms integrated into this layer ($UL = \{m_1, m_2, \dots, m_n\}$) would be orientated towards obtaining information on various levels, thereby making a distinction between independent tools which, potentially, could be integrated into a sole control panel:

- m_1: tool for visualizing the information captured by different sensors of the drones, as well as knowing their operational state by means of the proxy agents (PA).
- m_2: tool for visualizing the results of the analysis carried out by the cognitive layer agents (CL).
- m_3: tool for the human operators to access the recommendations suggested for the decision support module (DSM).

In the context of using the monitoring and simulation tools, it must be stressed that the tests carried out in the sports complex have been done by the SITL simulator. Apart from enabling the security code tests and experimental settings to be executed, the purpose of these type of tools is for practicing the use of the earth station from a desktop application. Essentially, SITL enables helicopters, airplanes or terrestrial vehicles to be simulated without needing any hardware. Moreover, SITL enables ArduPilot,[2] to be executed, and it is possible to access all the development tools available, such as interactive debuggers, static analyzers and tools for dynamic analysis.

As an example, Fig. 6 shows how the information looks that can be obtained directly in the event of using the SITL software. In particular, the screen shot shown is part of a mission carried out by a real drone (Iris + model) in the outskirts of a city considered in this case study. Due to legal matters in Spain and considering this drone does not exceed 25 kg, it is necessary to follow a series of rules. To be specific, the drone must operate away from crowded areas, buildings, people, cities, towns or inhabited places. Likewise, the drone cannot fly above 120 m, it must be within the visual range of the pilot and must not go more than 500 meters away. For these reasons, tests with a real drone had to be carried out on a plot of land far from built-up areas and in which there were no people or infrastructures nearby. This plot of land is shown in Fig. 6, which also includes the exit roads in the assigned mission.

Below, there is a basic definition of the surveillance component for monitoring the spread of toxic smoke considered in the case study of Section 4, using the formal model of normality analysis (Albusac et al., 2009) cited in previous sections.

5. Building a knowledge base to monitor toxic smoke propagation

The intelligent monitoring systems we have designed and developed in previous research works (Albusac et al., 2010) are made up of a set of surveillance components aimed at analyzing urban traffic scenarios. As a result of this experience, in this paper we have proposed the design of a new component, whose main objective is to analyze how toxic smoke spreads in an area monitored by UAVs, which is prone to undergoing a natural disaster (see Fig. 3 in order to recreate the situation graphically). The overall goal is to estimate the degree of risk there is in a scenario in which there is a fire with toxic smoke.

Just as presented in Section 3.3 and defined in Albusac et al. (2009), a surveillance component has been constructed which is based on a concept c_i (in this case study it is represented by the spread of toxic smoke). In reality, the surveillance components could be understood as being components of normality analysis (abbreviated as normality components), since their knowledge bases typically represent normal situations. Generally speaking, the number of normal situations is more restricted, and they are simpler to define than abnormal ones (Albusac et al., 2010).

Defining the concept C_i involves specifying a tuple made up of four elements $c_i = \langle V_i, DDV_i, KB_i, ie_i \rangle$. Here the following aspects must be stressed:

- A variable v_i can be input, if its values are useful for carrying out a normality analysis, or output, if its value is obtained as a result of carrying out the analysis itself. The values of a variable can be taken directly from a sensor installed in the UAV (e.g. presence) or may require processing beforehand (e.g. using a computer vision technique to find out how wide a smoke cloud is in respect to an image captured by a camera).
- A domain of definition DDV_i represents the set of valid values for a variable v_{ij}.
- The knowledge base KB_i can be defined on the basis of a set of normality restrictions Φ_i associated with the concept c_i. The normality analysis of a concept c_i would, here, depend on the degree of satisfaction of the restrictions defining it. The greater the degree of satisfaction of the restrictions, the higher the degree of normality in accordance with the c_i definition.

Using this definition model for normality components, the component for analyzing the spread of toxic smoke is made up of four input variables and one output variable.

The defined input variables are as follows:

[2] http://www.ardupilot.org.

Fig. 6. Deployment of the Iris+ drone in a real scenario, (a) View of the mission with the different exit roads. (b) information associated with the drone deployed when the mission is in progress.

- Distance with respect to the critical area, denoted as *distance*. This variable provides information about the distance there is between the area closest to the toxic smoke and the critical region it could potentially affect, such as, for example the periphery of a city.
- Propagation speed, denoted as *speed*. This variable shows the speed at which the fire or the cloud of toxic smoke spreads.
- Propagation direction with respect to the critical area, denoted as *direction*. Determines if the smoke spreads directly, partially or in the opposite direction to the critical region.
- Spread of fire/toxic smoke cloud, denoted as *spreading*. Shows the surface area taken up by the cloud of toxic smoke in the environment, captured by a surveillance camera.

The only output variable is the following:

- Degree of normality, denoted as *risk*. A high degree of normality implies a low or no risk in respect to the critical area under surveillance, whilst a low value shows the opposite.

The mathematical model used for defining this normality component is fuzzy logic (Zadeh, 1996), so the previously defined variables are fuzzy ones. Therefore, each domain of definition DDV_{ij} will be made up of a set of linguistic labels which are in turn linked to fuzzy sets. The values of each variable will determine the degree of membership to each fuzzy set. Fuzzy logic is suitable in this situation, due to the high degree of inaccuracy and vagueness there is in the underlying problem. In the upper part of Fig. 7 the specific set of linguistic labels for each of the variables is defined.

To give an example, and considering the instancing of this normality component in respect to the case study set out in Section 4 (in the step discussed in Section 4.2 to be specific), specifying the domains of definition of the variables *distance* and *speed* is considered for point p_6 in this case study. Specifically speaking, if the *distance* variable is associated with a domain of definition with the labels *very close*, *close*, *medium*, *far* and *very far*, and there is an intention to represent the distance between one point of interest, such as the city and the point at which a cloud of toxic smoke is detected, then the fuzzy sets shown in Fig. 8 could be used. This instancing must be extrapolated to the other domains of definition associated with the set of concepts or events of interest C_i to be monitored, since the range of values will depend on the environment which is to be analyzed.

When the four input variables have a new value, a fuzzification process begins, in which the degree of membership to the fuzzy sets of the domain of definition DDV_i of each variable V_i is studied. In this way, the numerical values are replaced with linguistic labels and degrees of membership. As discussed later, the set of restrictions Φ_i can be defined as a set of rules made up of linguistic labels. Moreover, these rules come with a fuzzy inference mechanism which determines which rules are activated and to what degree. The fuzzy inference mechanism is a possible alternative for defining the inference engine ie_i of a concept c_i. The degree of satisfaction of the rules or restrictions is determined by how much risk there is in a certain situation.

To obtain the value of the four input variables, in this research it has been assumed that the proxy agents PA would have a series of pre-processing modules which would analyze the images with computer vision techniques. The events generated by the PA are used as input for the information analysis agents IAA, which use the concept c_i to carry out the normality analysis.

Additionally, the strategy used for defining normal situations when defining a normality component can be supplemented by adding a minimum set of abnormal situations. This explicitly enables the most

Domain Definition Variables DDV_i

$DDV_{distance}$ = { very close, close, medium, far, very far}
DDV_{speed} = {very slow, slow, moderate, fast, very fast}
$DDV_{direction}$ = {straight to, partially to, opposite way}
$DDV_{spreading}$ = {small, medium, large}
DDV_{risk} = {very high, high, moderate, low}

Simplified set of rules that represent normality (situations with low or moderate risk)

R_{i-1}: IF Spreading is {small} THEN Risk IS {low}
R_i: IF Spreading is {medium} AND Distance is {very far} AND Direction is {partially to, opposite way} THEN Risk is {normal}
R_{i+1}: IF Spreading is {medium} AND Distance is {very far, far} AND Speed is {very slow, slow} AND Direction is {straigh to, partially to} THEN Risk is {moderate}$

Simplified set of rules that represent abnormality (situations with high or very high risk)

R_{j-1}: IF Spreading is {large} THEN Risk is {very high}
R_j: IF Speed is {fast, very fast} AND Direction is {straigh to} AND Spreading is {medium, large} THEN Risk is {very high}
R_{j+1}: IF Distance is {medium} AND Speed is {moderate} AND Direction is {straigh to, partially to} AND Spreading is {large} THEN Risk is {high}

Fig. 7. Definition of the domains of definition of the designed variables DDV_i and of the rules or restrictions which make up the knowledge base KB_i of a iaa_j.

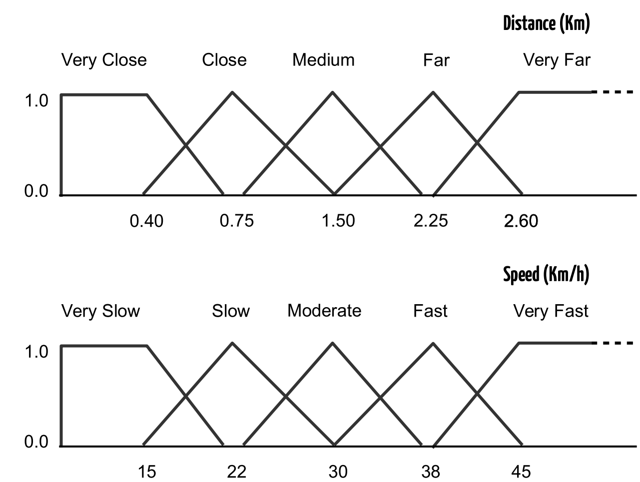

Fig. 8. Specification of the domains of definition of variables for the variables *distance* and *speed* for point p_6 of the case study envisaged in this research.

typical abnormal situations to be monitored. In this context, one of any of the following three situations may arise:

- A normal situation arises. In this event the system will detect it and will not suggest any course of action for the security staff to take.
- A known abnormal situation arises. The system would detect it and would suggest courses of action for the security staff to take.
- An unknown abnormal situation arises. This situation occurs when none of the rules are activated that are defined in the knowledge base KB_i of the concept c_i associated with iaa_j. That is, none of the defined restrictions are satisfied. In this situation the system is aware that something unusual is occurring and may communicate this and recommend courses of action to take in generally suspicious situations.

With this strategy in mind, Fig. 7 shows a simplified set of restrictions or rules which define low or moderate risk situations (normality with respect to the potential spread of toxic smoke) and high-risk situations (abnormality). It must be stressed that in a real deployment carried out by a commercial system, this set of rules would be necessary for completing it both in terms of normality and as regards probable abnormal situations.

Specifically speaking, the set of restrictions Φ_i of the normality analysis component of the spread of toxic smoke has been designed as a set of IF-THEN type fuzzy rules. The rules antecedent can be made up of four input variables *distance*, *speed*, *direction* and *spreading*, whilst the consequent will be associated with the output variable *risk*.

Just as can be seen in Fig. 7, these types of rules show a high degree of interpretability. They can easily be defined on the basis of reference situations (see Fig. 9), and they also make it very simple to modify the knowledge base KB_i. Moreover, each of these rules covers a multiple range of values for the input variables; so, it will just be necessary to define a reduced set of rules to cover most situations to be monitored.

6. Conclusions and future work

In this research we have presented multi-agent architecture which is used to deploy multi-agent systems aimed at retrieving and analyzing information by using mobile robots, which are usually UAVs, and for providing support to the decision-making process in environments affected by natural disasters. The architecture has been structured into layers, provides communication and cooperation mechanisms among agents and envisages defining a set of agents with different roles. Among these, most noteworthy are the information analysis agents, which are responsible for carrying out any analysis of what is taking place in the environment which is being monitored according to a set of concepts or events of interest.

When formalizing this architecture, there has been a discussion on the steps to take to deploy a multi-agent system which enables environments to be effectively analyzed, that contemplates aspects such as the environment characteristics, the specifications of the mobile robots, the definition of the concepts or events of interest to be analyzed, and the use of information fusion techniques to support decision-making. In this respect, the full cycle is covered from information retrieval to supporting decision-making. This discussion has been supplemented with a case study in which there is a hypothetical earthquake which affects an industrial complex situated near to a city. The environment has been analyzed on the basis of a concept or event of interest from the point of view of monitoring: the spread of clouds of toxic smoke. In this article, to monitor these potential risks, a normality component has been defined, and for this purpose a formal model has been used for normality analysis (Albusac et al., 2009) based on fuzzy logic which

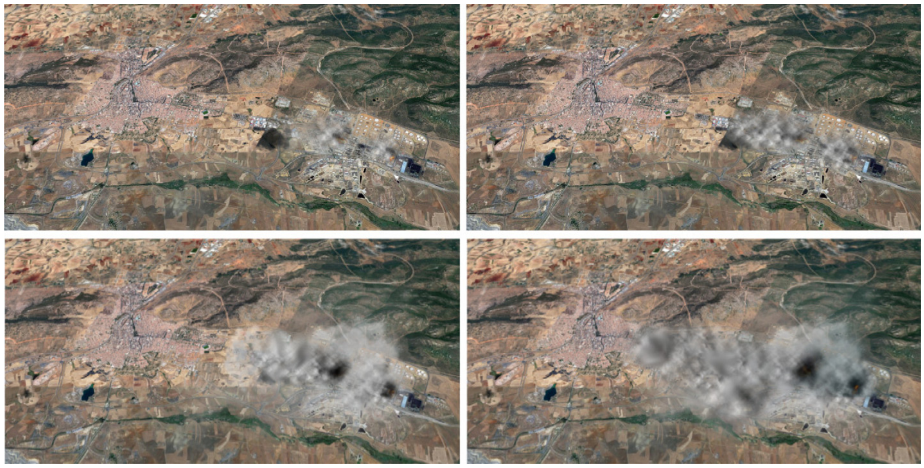

Fig. 9. Aerial view of the city (left) and industrial complex (right) affected by an earthquake which gives rise to a range of fires. Evolution of the cloud of toxic smoke as time passes simulated at four different instances.

has previously been used for monitoring mobile objects in urban traffic environments (Albusac et al., 2010). The aim is to generalize this component in order to monitor the spread of fire, an event closely linked to the spread of smoke.

Additionally, the systematic requirements for this type of architecture, and which have been presented in the Section 3.1 have been fulfilled. As regards availability, the events channels EC makes the information publishers and subscribers independent from each other. In terms of evolvability, this architecture has been designed to make the hardware components independent from the software. Therefore, the proxy agents PA provide the abstraction required to enable the remaining architecture agents to operate independently from the hardware devices used to obtain information about the environment. The integration requirement, which is very directly linked with evolvability, has likewise been justified by the design of the information retrieval layer, which contains the PA which are responsible for controlling the physical robots or drones integrated. As regards manageability, the architecture envisages using monitoring tools (see Figs. 2 and 6) which are fed from the knowledge created by the different agents of the architecture, especially from those integrated into the cognitive layer. For scalability, the devices for obtaining information about the environment make it possible to integrate new drones or terrestrial robots, on a communicative level, the architecture envisages defining the events channels EC which make the information publishers independent from the subscribers and the knowledge bases make it possible to include the analysis of new events of interest or threats. A large amount of the high-level agents incorporate knowledge bases used for analyzing the information obtained about the environment into their definition. For security requirements, a scene has been considered in which all information is encrypted, particularly that which is wireless.

Finally, as the main line of future research, we envisage improving and adapting the architecture by considering terrorist attack scenarios, which are usually characterized by their unpredictability and the chaos they cause. In a way, they resemble environments affected by natural disasters. The general objective would be to provide support in order to minimize the damage caused by the attack and, to once again, use a set of UVAs to gather useful information as fast as possible which could be useful to human staff who are responsible for responding to this situation. In this context, new agents would need to be included which are independent enough to retrieve information from other interesting data sources, such as, for example, social networks, which usually provide information which is almost in real time when an event with these characteristics occurs.

Acknowledgments

This work has been funded by the University of Castilla-La Mancha, Spain (Research Project GI20174147), the Department of Information Technologies and Systems, Spain (Ref. ID 00421372), and the Institute of Information Technologies and Systems, Spain (Ref. ID 00541R174).

References

Adams, S.M., Friedland, C.J., 2011. A survey of unmanned aerial vehicle (UAV) usage for imagery collection in disaster research and management. In: Proceedings of the Ninth International Workshop on Remote Sensing for Disaster Response. Stanford, CA, USA.

Albusac, J., Vallejo, D., Castro-Schez, J.J., Remagnino, P., Gonzalez, C., Jimenez, L., 2010. Monitoring complex environments using a knowledge-driven approach based on intelligent agents. IEEE Intell. Syst. 25 (3), 24–31.

Albusac, J., Vallejo, D., Jimenez-Linares, L., Castro-Schez, J.J., Rodriguez-Benitez, L., 2009. Intelligent surveillance based on normality analysis to detect abnormal behaviors. Int. J. Pattern Recognit. Artif. Intell. 23 (07), 1223–1244.

Andriluka, M., Schnitzspan, P., Meyer, J., Kohlbrecher, S., Petersen, K., Von Stryk, O., Roth, S., Schiele, B., 2010. Vision based victim detection from unmanned aerial vehicles. In: Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on. IEEE, pp. 1740–1747.

Arkin, R.C., 1998. Behavior-Based Robotics. MIT Press.

Arokiasami, W.A., Vadakkepat, P., Tan, K.C., Srinivasan, D., 2016. Interoperable multi-agent framework for unmanned aerial/ground vehicles: towards robot autonomy. Complex Intell. Syst. 2 (1), 45–59.

Baiocchi, V., Dominici, D., Milone, M.V., Mormile, M., 2013. Development of a software to plan UAVs stereoscopic flight: An application on post earthquake scenario in L'Aquila city. In: International Conference on Computational Science and Its Applications. Springer, pp. 150–165.

Beard, R.W., McLain, T.W., 2003. Multiple UAV cooperative search under collision avoidance and limited range communication constraints. In : Decision and Control, 2003. Proceedings. 42nd IEEE Conference on, Vol. 1, pp. 25–30.

Bellifemine, F., Poggi, A., Rimassa, G., 1999. JADE–A FIPA-compliant agent framework. In: Proceedings of PAAM, Vol. 99, London, No. 97–108, pp. 33.

Burgard, W., Moors, M., Stachniss, C., Schneider, F.E., 2005. Coordinated multi-robot exploration. IEEE Trans. Robot. 21 (3), 376–386.

Cao, Y., Yu, W., Ren, W., Chen, G., 2012. An overview of recent progress in the study of distributed multi-agent coordination. IEEE Trans. Ind. inform. 9 (1), 427–438.

Capitán, J., Merino, L., Caballero, F., Ollero, A., 2011. Decentralized delayed-state information filter (DDSIF): A new approach for cooperative decentralized tracking. Robot. Auton. Syst. 59 (6), 376–388.

Casper, J., Murphy, R.R., 2003. Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center. IEEE Trans. Syst. Man Cybern. B 33 (3), 367–385.

Chou, T.-Y., Yeh, M.-L., Chen, Y.C., Chen, Y.H., 2010. Disaster monitoring and management by the unmanned aerial vehicle technology. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. 38 (7B), 137–142.

Claes, D., Robbel, P., Oliehoek, F.A., Tuyls, K., Hennes, D., van der Hoek, W., 2015. Effective approximations for multi-robot coordination in spatially distributed tasks. In: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems. International Foundation for Autonomous Agents and Multiagent Systems, pp. 881–890.

Dasgupta, P., 2008. A multiagent swarming system for distributed automatic target recognition using unmanned aerial vehicles. IEEE Trans. Syst. Man Cybern. A 38 (3), 549–563.

Dias, M.B., Zlot, R., Kalra, N., Stentz, A., 2006. Market-based multirobot coordination: A survey and analysis. Proc. IEEE 94 (7), 1257–1270.

Doherty, P., Kvarnstrom, J., Wzorek, M., Rudol, P., Heintz, F., Conte, G., 2015. HDRC3: A distributed hybriddeliberative/reactive architecture for unmanned aircraft systems. In: Handbook of Unmanned Aerial Vehicles. Springer, pp. 849–952.

Fikar, C., Gronalt, M., Hirsch, P., 2016. A decision support system for coordinated disaster relief distribution. Expert Syst. Appl. 57, 104–116.

Foundation for Intelligent Physical Agents FIPA, 2004. FIPA agent management specification. http://www.fipa.org/specs/fipa00023/index.html.

Fukuda, T., Nakagawa, S., Kawauchi, Y., Buss, M., 1988. Self organizing robots based on cell structures-CKBOT. In: Intelligent Robots, 1988., IEEE International Workshop on. IEEE, pp. 145–150.

Garcia-Molina, H., 1982. Elections in a distributed computing system. IEEE Trans. Comput. (1), 48–59.

Gaszczak, A., Breckon, T.P., Han, J., 2011. Real-time people and vehicle detection from UAV imagery. In: IS&T/SPIE Electronic Imaging. International Society for Optics and Photonics, p. 78780B.

Haghighi, R., Cheah, C.-C., 2012. Multi-group coordination control for robot swarms. Automatica 48 (10), 2526–2534.

Henning, M., 2004. A new approach to object-oriented middleware. IEEE Internet Comput. 8 (1), 66–75.

Hollinger, G.A., Singh, S., 2012. Multirobot coordination with periodic connectivity: Theory and experiments. IEEE Trans. Robot. 28 (4), 967–973.

Ingrand, F., Ghallab, M., 2017. Deliberation for autonomous robots: A survey. Artificial Intelligence 247, 10–44.

Jennings, N.R., Sycara, K., Wooldridge, M., 1998. A roadmap of agent research and development. Auton. agents multi-agent syst. 1 (1), 7–38.

Ji, X., Wang, X., Niu, Y., Shen, L., 2015. Cooperative search by multiple unmanned aerial vehicles in a nonconvex environment. Math. Probl. Eng. 2015, 1–19.

Kurnaz, S., Cetin, O., Kaynak, O., 2010. Adaptive neuro-fuzzy inference system based autonomous flight control of unmanned air vehicles. Expert Syst. Appl. 37 (2), 1229–1234.

Mataric, M.J., 1997. Behaviour-based control: Examples from navigation, learning, and group behaviour. J. Exp. Theor. Artif. Intell. 9 (2–3), 323–336.

Moon, S., Oh, E., Shim, D.H., 2013. An integral framework of task assignment and path planning for multiple unmanned aerial vehicles in dynamic environments. J. Intell. Robot. Syst. 70 (1–4), 303–313.

Murphy, R.R., 2004. Human-robot interaction in rescue robotics. IEEE Trans. Syst. Man Cybern. C 34 (2), 138–153.

Nagatani, K., Kiribayashi, S., Okada, Y., et al., 2013. Emergency response to the nuclear accident at the Fukushima Daiichi Nuclear Power Plants using mobile rescue robots. J. Field Robotics 30 (1), 44–63.

Parker, L.E., 1998. ALLIANCE: An architecture for fault tolerant multirobot cooperation. IEEE Trans. Robot. Autom. 14 (2), 220–240.

Portmann, J., Lynen, S., Chli, M., Siegwart, R., 2014. People detection and tracking from aerial thermal views. In: 2014 IEEE International Conference on Robotics and Automation, ICRA. IEEE, pp. 1794–1800.

Portugal, D., Rocha, R.P., 2013a. Multi-robot patrolling algorithms: examining performance and scalability. Adv. Robot. 27 (5), 325–336.

Portugal, D., Rocha, R.P., 2013b. Distributed multi-robot patrol: A scalable and fault-tolerant framework. Robot. Auton. Syst. 61 (12), 1572–1587.

Power, D.J., Sharda, R., Burstein, F., 2015. Decision Support Systems. Wiley Online Library.

Rao, A.S., Georgeff, M.P., 1991. Modeling rational agents within a BDI-architecture. In: KR, Vol. 91, pp. 473–484.

Sislak, D., Volf, P., Komenda, A., Samek, J., Pechoucek, M., 2007. Agent-based multi-layer collision avoidance to unmanned aerial vehicles. In: 2007 International Conference on Integration of Knowledge Intensive Multi-Agent Systems. IEEE, pp. 365–370.

Sun, T.-Y., Huo, C.-L., Tsai, S.-J., Yu, Y.-H., Liu, C.-C., 2011. Intelligent flight task algorithm for unmanned aerial vehicle. Expert Syst. Appl. 38 (8), 10036–10048.

Torres, M., Pelta, D.A., Verdegay, J.L., Torres, J.C., 2016. Coverage path planning with unmanned aerial vehicles for 3D terrain reconstruction. Expert Syst. Appl. 55, 441–451.

Wang, Y., de Silva, C.W., 2008. A machine-learning approach to multi-robot coordination. Eng. Appl. Artif. Intell. 21 (3), 470–484.

Weiss, G., 1999. Multiagent Systems: a Modern Approach to Distributed Artificial Intelligence. MIT press.

Wooldridge, M.J., 2001. Multi-Agent Systems: an Introduction. Wiley Chichester.

Wooldridge, M., Jennings, N.R., 1995. Intelligent agents: Theory and practice. knowl. eng. rev. 10 (02), 115–152.

Xu, Z., Yang, J., Peng, C., Wu, Y., Jiang, X., Li, R., Zheng, Y., Gao, Y., Liu, S., Tian, B., 2014. Development of an UAS for post-earthquake disaster surveying and its application in Ms7. 0 Lushan Earthquake, Sichuan, China. Comput. Geosci. 68, 22–30.

Yan, Z., Jouandeau, N., Cherif, A.A., 2013. A survey and analysis of multi-robot coordination. Int. J. Adv. Robot. Syst. 10.

Zadeh, L., 1996. Fuzzy logic= computing with words. IEEE Trans. Fuzzy Syst. 4 (2), 103–111.

Zhao, Y., Wang, X., Wang, C., Cong, Y., Shen, L., 2019. Systemic design of distributed multi-UAV cooperative decision-making for multi-target tracking. Auton. Agents Multi-Agent Syst. 33 (1–2), 132–158.