Journal of Intelligent & Robotic Systems manuscript No.
(will be inserted by the editor)

Experimental Results in Multi-UAV Coordination for Disaster Management and Civil Security Applications

Iván Maza · Fernando Caballero
Jesús Capitán · J.R. Martínez-de-Dios
Aníbal Ollero

Received: date / Accepted: date

Abstract This paper describes a multi-UAV distributed decisional architecture developed in the framework of the AWARE Project together with a set of tests with real Unmanned Aerial Vehicles (UAVs) and Wireless Sensor Networks (WSNs) to validate this approach in disaster management and civil security applications.

The paper presents the different components of the AWARE platform and the scenario in which the multi-UAV missions were carried out. The missions described in this paper include surveillance with multiple UAVs, sensor deployment and fire threat confirmation. In order to avoid redundancies, instead of describing the operation of the full architecture for every mission, only non-overlapping aspects are highlighted in each one. Key issues in multi-UAV systems such as distributed task allocation, conflict resolution and plan refining are solved in the execution of the missions.

Keywords Multi-UAV · Distributed Decision Making · Coordination · Cooperation

This work was partially supported by the AWARE project (IST-2006-33579) funded by the European Commission and the ROBAIR Project funded by the Spanish Research and Development Program (DPI2008-03847). J.R. Martínez-de-Dios also thanks the support provided by the CONET Network of Excellence (INFSO-ICT-224053) funded by the European Commission

I. Maza, F. Caballero, J. Capitán, J.R. Martínez-de-Dios and A. Ollero
Universidad de Sevilla, Escuela Superior de Ingenieros, Camino de los Descubrimientos s/n, 41092 Sevilla (Spain)
Tel.: +34-954-487361
Fax: +34-954-487340
E-mail: imaza@us.es, fcaballero@us.es, jescap@cartuja.us.es, jdedios@cartuja.us.es, aollero@cartuja.us.es

A. Ollero
Center for Advanced Aerospace Technology (CATEC), Parque Tecnológico y Aeronáutico de Andalucía, C. Wilbur y Orville Wright 17-19-21, 41309, La Rinconada (Spain)
E-mail: aollero@catec.aero

The original publication is available at www.springerlink.com in this link:
http://dx.doi.org/10.1007/s10846-010-9497-5

1 Introduction

This paper describes a distributed architecture for the autonomous coordination and cooperation of multiple Unmanned Aerial Vehicles (UAVs) [21]. In this cooperation each individual executes a set of tasks: Subgoals that are necessary for achieving the overall goal of the system, and that can be achieved independently of other subgoals. These tasks are allocated to perform a given mission in an efficient manner according to planning strategies [8]. The robots cooperate explicitly and with purpose, and then, this cooperation is defined as intentional cooperation [18].

Key issues in these systems include determining which robot should perform each task (task allocation) in order to maximize the efficiency of the team and ensuring the proper coordination among team members to complete their mission [19]. The multi-robot task allocation problem requires defining some metrics to assess the relevance of assigning given tasks to particular robots. A domain independent taxonomy of this problem is presented in [7]. In the last years, a popular approach to solve the multi-robot task allocation problem in a distributed way is the application of market-based negotiation rules. Usual implementations of those rules [5,1,6] are based on the Contract Net Protocol [20]. In those approaches, the messages coming from the cooperating robots are those involved in the negotiation process: announce a task, bid for a task, allocate a task, ask for the negotiation token, etc.

Once the tasks have been allocated, it is necessary to coordinate the motions of the vehicles, which can be done by means of suitable multi-vehicle path/velocity planning strategies. Even if the vehicles are explicitly cooperating through messages, a key element in many approaches is the updated information about the state of the neighbors.

In general terms, teams composed by heterogeneous members involve challenging aspects, even for the intentional cooperation approach. In [17,16] the current state of the technology, existing problems and potentialities of platforms with multiple heterogeneous UAVs is studied. This heterogeneity is two-fold: firstly in the UAV platforms looking to exploit the complementarities of different aerial vehicles, such as helicopters and airships, and secondly in autonomy, ranging from pure remotely teleoperated vehicles to fully autonomous aerial robots.

On the other hand, cooperative perception is another challenging issue in multi-robot systems and can be defined as the task of creating and maintaining a consistent view of a world containing dynamic objects by a group of robots each equipped with one or more sensors. A team of vehicles can simultaneously collect information from multiple locations and exploit the information derived from multiple disparate points to build models that can be used to take decisions. In [15] cooperation perception methods for multi-UAV systems are proposed. Each UAV extracts knowledge, by applying individual perception techniques [12], and the overall cooperative perception is performed by merging the individual results. This approach requires knowing the relative position and orientation of the UAVs. In many outdoor applications it is assumed that the position of all the robots can be obtained by means of GPS and broadcasted through the communication system. However, if this is not the case, the robots should be capable of identifying and localizing each other [11] which could be difficult with the on-board sensors. Another approach consists on identifying common objects in the scene. Then, under certain assumptions, the relative pose displacement between the vehicles can be computed from these correspondences. This strategy has been described with heterogeneous UAVs in [9].

The original publication is available at www.springerlink.com in this link:
http://dx.doi.org/10.1007/s10846-010-9497-5

The multi-UAV coordination and control architecture developed in the COMETS Project was demonstrated for the autonomous detection and monitoring of forest fires [17] by using two helicopters and one airship. Regarding teams involving aerial and ground vehicles, the CROMAT Project architecture also implemented cooperative perception and multi-robot task allocation techniques [24] that were demonstrated in fire detection, monitoring and extinguishing missions.

In [13], the AWARE Project [1] distributed architecture for the autonomous coordination and cooperation of multiple UAVs for civil applications was presented. That paper described the development stage of the AWARE architecture in 2008 and a preliminar field experiment. The validation process of the architecture has continued during 2009 with more field experiments involving up to four autonomous helicopters. The validation included the following multi-UAV missions for civil applications:

- Multi-UAV cooperative area surveillance.
- Wireless sensor deployment.
- Fire threat confirmation and extinguishing.
- Load transportation and deployment with single and multiple UAVs.
- People tracking.

This paper presents some of the above missions performed with real UAVs in the experiments of the AWARE Project carried out in 2009. The paper is structured as follows: First, Section 2 describes the main components of the distributed architecture for multi-UAV cooperation developed in the AWARE Project. Section 3 presents the task model used in the experiments. Later, the scenario and main actors in the tests are described in Sect. 4, whereas Section 5 presents the multi-UAV missions themselves. Finally, Section 6 closes the paper with the conclusions.

2 Distributed Decisional Architecture of the AWARE Platform

A global mission for the AWARE platform is specified by the operator using the Human Machine Interface (HMI) software applications. Each mission $\mathcal{M}$ consists in a set of *tasks* (possibly ordered) that should be executed by the platform. The distribution of the tasks among the different UAVs (task allocation process) could be manually done by the user or may be autonomously performed in a distributed manner. The latter might be mandatory in situations where large numbers of UAVs have to interact, where direct communication with a central station is not possible, or where the local dynamics of the situation require timely reaction of the UAVs involved on it.

The tasks that will be executed by the AWARE platform involve coordination, mainly for sharing space, and cooperation, for example for the surveillance at different altitudes or from different viewpoints of the same object, or when an UAV plays the role of a radio re-transmitter from/to other UAVs and the central station. Cooperation includes coordination, but there is role sharing between the subsystems to achieve a global common task.

It is worth to mention that the main objective in the design of the multi-UAV architecture was to impose few requirements to the execution capabilities of the autonomous vehicles to be integrated in the platform. Basically, those vehicles should be able to move to a given location and activate their on-board instruments when required. Then,

[1] http://www.aware-project.net

The original publication is available at www.springerlink.com in this link:
http://dx.doi.org/10.1007/s10846-010-9497-5

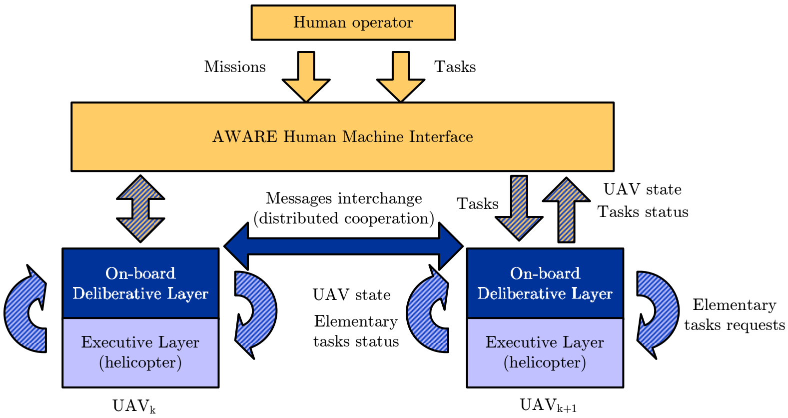


Fig. 1 Global overview of the distributed multi-UAV system architecture.

autonomous vehicles from different manufacturers and research groups can be easily integrated in the AWARE architecture.

The global picture of the AWARE distributed UAV system is shown in Fig. 1. In each UAV, there are two main layers: the On-board Deliberative Layer (ODL) and the proprietary Executive Layer (EL). The former deals with the high-level distributed decision-making mentioned above, whereas the latter is in charge of the execution of elementary tasks (see Sect. 3). In the interface between both layers, the ODL sends elementary task requests and receives the execution state of each elementary task and the UAV state. For distributed decision-making purposes, interactions among the ODLs of different UAVs are required. Finally, the HMI software allows the user to specify the missions and tasks to be executed by the platform, and also to monitor the execution state of the tasks and the status of the different UAVs.

A more detailed view of the internal ODL architecture is shown in Fig. 2. As it has been mentioned above, the ODL has interactions with its executive layer and with the ODLs of other UAVs as well as with the HMI. The different modules shown in the ODL supports the distributed decision-making process involving cooperation and coordination. Further details about the operation of each ODL module can be found in [13]. Nevertheless, the role of the modules involved in the missions described in this paper will be detailed in Sect. 5.

Next section summarizes the final task model adopted in the interface among ODLs and between the ODL and its EL. It updates the task model presented in [13] used in 2008 in the AWARE platform.

3 Task Model

Let us consider a mission $\mathcal{M}$ specified by the AWARE platform user. This mission is decomposed (autonomously or manually) in a set of partially ordered tasks $\mathcal{T}$. Let us define a task with unique identifier k and type λ allocated to the i-th UAV as $\tau_i^k = (\lambda, {}^{-}\Omega, \Omega^{+}, \varepsilon, \Pi)$, where ${}^{-}\Omega$ and Ω^{+} are respectively the set of preconditions and postconditions of the task, and ε is the state associated to the task evolution (see

The original publication is available at www.springerlink.com in this link:
http://dx.doi.org/10.1007/s10846-010-9497-5

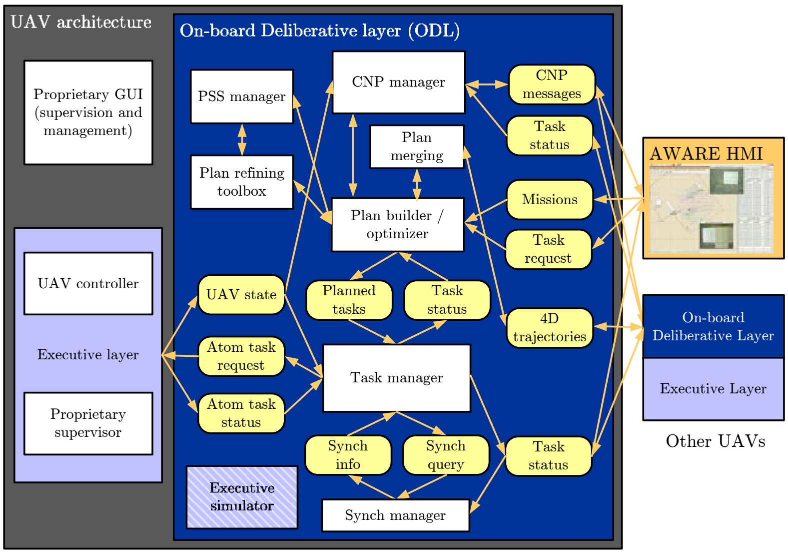


Fig. 2 Detailed view of the internal On-board Deliberative Layer (ODL) architecture for a single UAV.

Table 1 Possible states considered in the status evolution of a task τ_i^k.

State (ε)	Description
`EMPTY`	No task
`SCHEDULED`	The task is waiting to be executed
`RUNNING`	The task is in execution
`CHECKING`	The task is being checked against inconsistencies and static obstacles
`MERGING`	The task is in the plan merging process to avoid conflicts with the trajectories of other UAVs
`ABORTING`	The task is in process to be aborted. If it is finally aborted, the status will change to `ABORTED`, and otherwise will return to `RUNNING`
`ABORTED`	The task has been aborted (the human operator has aborted it or the UAV was not able to accomplish the task properly)
`ENDED`	The task has been accomplished properly

Table 1). Finally, $\Pi = \{\pi^1, \pi^2, \ldots, \pi^m\}$ is the set of m parameters that characterizes the task. The parameter set for some tasks will be explained later in Sect. 5.

Regarding the type of task (λ) at the ODL level of the architecture, the list shown in Table 2 has been considered in the AWARE platform.

Preconditions and postconditions are event-based mechanisms that can deal with events related to the evolution of the tasks states (see Table 1), as well as to the reception of messages, detection of a given event by the perception system, elapsing of a certain time period, etc. Then, the execution of a task starts when all the associated preconditions are satisfied. On the other hand, it is also possible to specify postconditions, i.e. conditions which satisfaction triggers the abortion of a task. If a task does not have any precondition or postcondition, then $\tau_i^k = (\lambda,^- \Omega = \emptyset, \Omega^+ = \emptyset, \varepsilon, \Pi)$.

The original publication is available at www.springerlink.com in this link:
http://dx.doi.org/10.1007/s10846-010-9497-5

Table 2 Type of tasks (λ) considered at the ODL level.

Type of task (λ)	Description
`TAKE-OFF`	The UAV takes off and stabilizes at a default safe altitude, then switches to the wait mode, waiting for further instructions
`LAND`	The UAV starts landing procedures, lands, and is set to a ground safe mode
`GOTO`	The UAV moves from its current location to a point P (or to its vicinity)
`GOTOLIST`	The UAV moves from its current location to each of the points of the waypoints list, following the order of the points
`DEPLOY`	The UAV moves from its current location to a point P (or to its vicinity) and activates its payload in order to deploy a device
`TAKE-SHOT`	The UAV moves from its current location to a point P (or to its vicinity) in order to take images of a given location L. P is computed to have L in the center of the on-board camera's field of view
`WAIT`	The UAV is set to a safe waiting mode: hover or pseudo-hover, during a given period
`SURV`	The UAV covers a given area defined by a polygon at a certain altitude
`TRACK`	The perception system of the UAV starts to operate in tracking mode, providing (if possible) location estimations of a given object (fire, persons, etc.). The UAV moves to a location that allows to improve the estimation of the location using the sensors on-board
`HOME`	The UAV is commanded to return home

An example of a precondition or a postcondition related to the evolution of the tasks is the "end of task" event of a different task. Furthermore, thanks to the synchronization manager module (see Fig. 2), it is possible to specify preconditions between tasks of different UAVs. Finally, it should be mentioned that perception events (not related to the execution of a task) such as the detection of a fire or a fireman in a disaster scenario, could be also the precondition or postcondition for a task (i.e. a tracking task).

The ODL processes the tasks received and generates simpler tasks, called elementary tasks, that are finally sent to the executive layer of the UAV. Let us define an elementary task with unique identifier k and type $\hat{\lambda}$ allocated to the i-th UAV as $\hat{\tau}_i^k = (\hat{\lambda}, \hat{\Pi}, \hat{\varepsilon})$ where $\hat{\Pi} = \{\hat{\pi}^1, \hat{\pi}^2, \ldots, \hat{\pi}^{\hat{m}}\}$ is the set of $\hat{m}$ parameters which characterizes the elementary task and $\hat{\varepsilon}$ is the state associated to the elementary task evolution. `RUNNING` and `ENDED` are the only states considered for the elementary tasks, simplifying the design of the executive layer software on-board the UAVs.

Then, the vehicles to be integrated in the AWARE platform should be able to receive elementary tasks, report their associated execution states and execute them. A small set of elementary tasks have been considered in order to allow the integration of a broader number of vehicles from different manufacturers and research groups. Basically, those vehicles should be able to move to a given location and activate their on-board instrument when required. Additionally, autonomous take-off and landing capabilities are also required (see Table 3). On the other hand, as an example, Table 4 shows the seven parameters that are considered in the elementary `GOTO` task.

The original publication is available at www.springerlink.com in this link:
http://dx.doi.org/10.1007/s10846-010-9497-5

Table 3 Type of elementary tasks ($\hat{\lambda}$) considered at the Executive Layer (EL). It can be seen as a subset of the tasks considered at the ODL level (see Table 2).

Type of task ($\hat{\lambda}^k$)	Description
`TAKE-OFF`	The UAV takes off and stabilizes at a default safe height, then switches to the wait mode, waiting for further instructions.
`GOTO`	The UAV moves from its current location to a point P (or to its vicinity) and activates its on-board instrument if required.
`LAND`	The UAV starts landing procedures, lands, and is set to a ground safe mode.

Table 4 Elementary task with type $\hat{\lambda}$ = GOTO: list of parameters.

Parameters ($\hat{\Pi}$)	Description
$\hat{\pi}^1$(x)	East UTM coordinate (m)
$\hat{\pi}^2$(y)	North UTM coordinate (m)
$\hat{\pi}^3$(Altitude)	Altitude (m) ellipsoid-based datum WGS84
$\hat{\pi}^4$ (Speed)	Desired speed (m/s) along the path to the waypoint
$\hat{\pi}^5$(ForceHeading)	1: Use the specified heading for the flight, 0: Heading fixed by the EL
$\hat{\pi}^6$(Heading)	Heading (degree) for the flight (N is 0°, E is 90°, W is -90° and S is 180°)
$\hat{\pi}^7$(Payload)	1: to activate the payload around the location of the waypoint, 0: not to activate

Next section describes the experimentation scenario used in the AWARE Project to validate the architecture and task model previously presented.

4 Experimentation Scenario

Three general experiments sessions, one per year of the AWARE Project, were conducted for platform integration and validation purposes in a common scenario. These experiments involved UAVs, a wireless ground sensor network with static and mobile nodes, ground cameras and fire trucks. This common scenario was settled in the facilities of the Protec-Fire company (Iturri group) (see Fig. 3) in Utrera (Spain). The AWARE experiments offered the framework to test the distributed implementation of the architecture previously described in Sect. 2.

Figure 3(a) shows the structure used to simulate a building where an emergency could be declared. In the structure there were several nodes of the Wireless Sensor Network (WSN) equipped with different types of sensors (temperature, humidity, CO, smoke, etc.) that could generate an alarm if a potential fire was detected. The fire in the building was simulated using both fire and smoke machines.

In the surroundings of the building, the following elements were present (see Fig. 3(a)):

- A set of static WSN nodes deployed on the ground.
- Several barrels close to the building. The fire declared in the building could propagate to its surroundings and reach other infrastructures with harmful consequences. Then, the barrels were intended to simulate those infrastructures around the building.

The original publication is available at www.springerlink.com in this link:
http://dx.doi.org/10.1007/s10846-010-9497-5

(a) Structure used to simulate a building

(b) Tents used by the AWARE team

Fig. 3 Common scenario settled in the facilities of the Protec-Fire company (Iturri group) in Utrera (Spain).

- Fixed cameras mounted on tripods. There were two visual cameras in the area around the building to monitor the scenario.
- A fire machine for outdoors used to simulate a possible propagation of the fire in the building to other infrastructures.
- Several dummy bodies were used to simulate victims in the building and also on the ground.

Next paragraphs briefly describe the main hardware components used in the tests.

4.1 Unmanned Aerial Vehicles (UAV)

A total of five small-scale autonomous helicopters were available in the third year of the AWARE project:

- Four *TUB-H* helicopters (see Fig. 4) developed by the Technische Universität Berlin (TUB).
- One *FC III E SARAH* helicopter (Electric Special Aerial Response Autonomous Helicopter) developed by the Flying-Cam (FC) company (see Fig. 5). The Technische Universität Berlin also cooperated with Flying-Cam for the operation of the prototype in autonomous flight.

The *TUB-H* UAVs were ready to mount different types of payloads depending on the particular mission thanks to a mechanical design based on a frame composed of strut profiles. Through the use of these profiles, the location of hardware components can be altered and new hardware can be installed easily. This allowed quick reconfiguration of the UAVs for different applications, easy replacement of defective hardware and alteration of the position of different components to adjust the UAVs centre of gravity. Then, the following payloads were used during the different missions:

- Fixed visual and infrared cameras (see Fig. 6).
- The Node Deployment Device (NDD) developed by the Technische Universität Berlin (see Fig. 7(a)). This system allowed to attach several sensors to the helicopter and sequentially drop them into the desired position with an error below one meter.
- The Load Transportation Device (LTD) developed by the Technische Universität Berlin (see Fig. 7(b)). This device allowed the transportation of different objects

The original publication is available at www.springerlink.com in this link:
http://dx.doi.org/10.1007/s10846-010-9497-5

(a) Three *TUB-H* helicopters on the landing pads (b) Detailed view of one *TUB-H* helicopter

Fig. 4 Fleet of *TUB-H* helicopters developed by the Technische Universität Berlin (TUB) used in the experiments. The fourth *TUB-H* unit was a ready-to-fly, spare helicopter.

Fig. 5 The *FC III E SARAH* helicopter developed by the Flying-Cam (FC) company.

attached to a rope. The LTD was composed of a two axis cardan joint with two magnetic encoders attached to each axis, and other sensors/actuators that enabled the UAV to measure the force and rope orientation relative to the fuselage, as well as to release the load in case of emergencies.

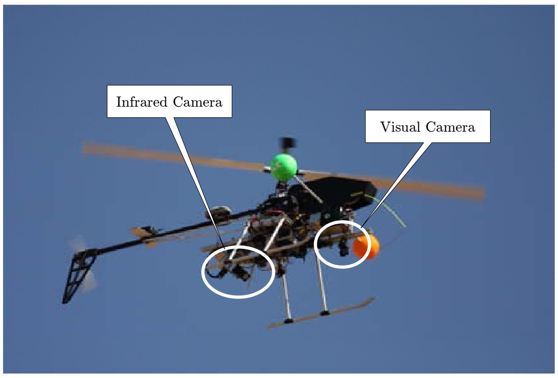
4.2 Ground Cameras (GC)

In the experimentation scenario, there were several fixed cameras on the ground that were intended to emulate a surveillance camera network in an urban setting. The system was based on firewire cameras connected to a PC104 that had a wireless link to the AWARE network (see Fig. 8).

The AWARE Perception System [4] was a distributed software application with the role of creating and maintaining a consistent view of a world containing dynamic objects. It was based on different software instances called Perception Subsystems (PSSs), that were associated with each sensor or group of sensors in the platform.

Then, on each GC PC104 there was a Perception Subsystem (PSS) application that processed the images and computed a local estimation of the states of the objects in the field of view. This local estimation was fused in a distributed perception system that integrated measurements from different information sources such as the visual and infrared cameras, as well as the wireless sensor network.

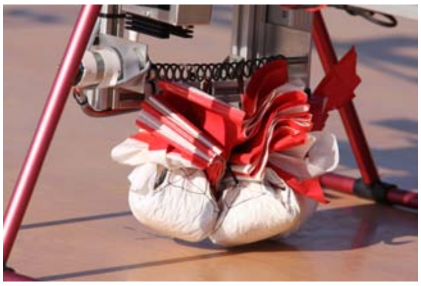
The original publication is available at www.springerlink.com in this link:
http://dx.doi.org/10.1007/s10846-010-9497-5



(a) Visual and infrared cameras on-board the *TUB-H* helicopter

(b) Visual camera on-board the *TUB-H* helicopter

Fig. 6 Visual and infrared cameras on-board the *TUB-H* helicopter. On the left photograph, both cameras are mounted on-board the helicopter with different orientation angles. The right photograph shows a detailed view of the visual camera with its analog transmitter.



(a) Node Deployment Device (NDD)

(b) *TUB-H* helicopter equipped with the LTD on the landing pad

Fig. 7 Node Deployment Device (NDD) and Load Transportation Device (LTD), both developed by the Technische Universität Berlin (TUB).

4.3 Wireless Sensor Network (WSN)

During the experiments, there were several wireless sensor networks deployed on the ground area in front of the building and also inside. Each WSN had a laptop acting as a gateway connected to the AWARE network through a wireless link. All the WSNs were measuring different variables such as temperature, humidity, CO, etc. in order to generate alarms if a potential fire were detected (see Fig. 9(a)). Additionally, some nodes were able to measure the Received Signal Strength Indication (RSSI) that allowed its gateway's PSS to provide estimations of the firemen's locations if they were equipped with the same type of nodes (see Figs. 9(b) and 9(c)). Those estimations are fused in a distributed manner with the measurements provided by the different cameras (ground and on-board the UAVs) to improve the results.

The sensors of the WSN can be also autonomously deployed when required by the UAVs thanks to the Node Deployment Device mentioned before (see Fig. 7(a)). The

The original publication is available at www.springerlink.com in this link:
http://dx.doi.org/10.1007/s10846-010-9497-5

Fig. 8 Ground cameras used in the experimentation scenario. Each firewire camera was connected to a PC104 with a wireless link to the AWARE network.

(a) Detail of a WSN node

(b) Node located in the building

(c) Node located on the ground in front of the building

Fig. 9 Detail of the WSN nodes used in the experiments. The type of node shown in the photograph (a) was used to measure different variables such as temperature, humidity, CO, etc. On the other hand, for the localization of the firemen in the area in front of the building (and also inside it) a different type of node (Xbow Mica2) able to measure the RSSI (shown in the photographs (b) and (c)) was used.

purpose of the deployment is twofold – firstly, allows to extend the area covered by the sensors, and secondly, can help to recover the connectivity in a WSN that has lost some nodes due to the difficult conditions in the scenario. Finally, if the transported sensor can measure the RSSI, it is also possible to apply different techniques [3, 2] to estimate the location of the WSN nodes before the deployment.

The original publication is available at www.springerlink.com in this link:
`http://dx.doi.org/10.1007/s10846-010-9497-5`

Table 5 Different missions with an indication of the subsystems involved in each one.

Mission	Brief description	HMI	UAV	GC	WSN	FT
0	Node deployment	✓	✓		✓	
1	Firemen tracking	✓	✓	✓	✓	
2	Firemen tracking	✓	✓	✓	✓	
3	Surveillance	✓	✓			
4	Node deployment & fire monitoring	✓	✓		✓	✓
5	Node deployment & fire monitoring	✓	✓		✓	✓
6	Fire monitoring	✓	✓		✓	✓
7	Surveillance	✓	✓			✓
8	Load transportation	✓	✓			
9	Node deployment	✓	✓		✓	
10	Fire monitoring	✓	✓	✓	✓	✓
11	Surveillance	✓	✓			✓

4.4 Fire Truck (FT)

In the experimentation scenario, there was a fire truck equipped with an automated water cannon. Then, once the AWARE platform has detected a fire in a given location, the water cannon could be commanded from the HMI to point and deliver water on that location. The system was equipped with a GPS and an IMU, and computed the required angles to deliver water on the intended place.

5 Multi-UAV Missions in the AWARE Project Experiments

During the AWARE Project experiments different multi-UAV missions were performed in order to validate the distributed software architecture. These missions included:

- Multi-UAV firemen tracking.
- UAV sensor deployment to extend the WSN coverage.
- UAV fire confirmation and extinguishing.
- Multi-UAV surveillance.
- Single and multi-UAV load transportation.

The missions could be also sequenced if the corresponding preconditions to start were satisfied. For instance, after the UAV sensor deployment, a mission for fire confirmation and extinguishing could start if a fire had been detected by the sensors deployed.

A summary of the different multi-UAV missions carried out in the AWARE experiments and the subsystems involved is shown in Table 5.

The integrated missions included coordinated flights involving sensor deployment, fire detection, monitoring and extinguishing, surveillance using two coordinated helicopters, tracking of firemen using two coordinated helicopters, load transportation using a single helicopter, and load transportation [13] using three coupled helicopters.

Although the range of missions in the project is wider, this paper is focused on two missions: sensor deployment and fire confirmation; and multi-UAV surveillance. For a better understanding of those missions, Table 6 shows links to the videos with the live execution during the AWARE Project experiments. The videos contain some fragments of the full mission and alternates views of the HMI screen along with the action of the helicopters from an external camera.

The original publication is available at www.springerlink.com in this link:
http://dx.doi.org/10.1007/s10846-010-9497-5

Table 6 Videos showing the live execution of the missions presented in this paper. The videos can be played using the VLC media player (`http://www.videolan.org`).

Mission	Link
Sensor deployment	`http://www.aware-project.net/videos/sens.avi`
Fire confirmation and extinguishing	`http://www.aware-project.net/videos/ir.avi`
Multi-UAV surveillance	`http://www.aware-project.net/videos/surv.avi`

5.1 Node Deployment and Fire Monitoring

This mission (identified as #5 in Table 5) was performed on 25th May 2009. The initial situation was as follows:

- A fire alarm had been declared in the building by the WSN inside. This fire had been also confirmed with the ground cameras outside the building.
- After a surveillance mission, several fuel barrels close to the building had been localized.

There were two UAVs ready to fly on the landing pads:

- UAV 1 equipped with an infrared camera aligned with the fuselage and pointing downwards 45°.
- UAV 2 equipped with the node deployment device charged with three sensors.

As there was risk of fire propagation from the building to the fuel barrels, a deployment mission was specified in order to place several sensors in the area between them at the locations of the waypoints `wp1`, `wp2` and `wp3` (see Fig. 11). Let us denote the corresponding tasks as τ^8, τ^9 and τ^{10} respectively.

The distributed negotiation process for the sensor deployment tasks started and the involved messages interchanged are shown in Fig. 10. The negotiation is based on the SIT algorithm [22, 23] that follows a market-based approach to solve the distributed task allocation problem. This algorithm was running during the tests on the CNP manager module of the internal ODL architecture shown in Fig. 2. The HMI application announced the three tasks and the two UAVs bid for them with their corresponding insertion costs. The cost of the tasks was computed using the distance to the waypoints as metric, and the insertion costs were the difference between the costs of the plans with and without the new tasks. It can be seen that the bids from UAV 1 were infinite because it was not equipped with the NDD.

When bidding, the plan builder module (see Fig. 2) checks different insertion points in the current plan in order to find the lowest associated cost (lowest bid).

All the tasks were initially allocated to UAV 2. According to the SIT algorithm, UAV 2 asked for the token to announce again the tasks won. The HMI application sent the token to UAV 2 and tasks τ^8, τ^9 and τ^{10} were announced again. All the bids received were infinite, so the tasks were definitely allocated to UAV 2: τ_2^8, τ_2^9 and τ_2^{10}.

Then, each deployment task was decomposed by the plan refiner module (see Fig. 2), leading to four elementary tasks:

1. Reach the waypoint.
2. Go down until the altitude is $h_d = 3.5$ meters above the ground.
3. Activate the NDD to deploy the sensor.

The original publication is available at www.springerlink.com in this link:
`http://dx.doi.org/10.1007/s10846-010-9497-5`

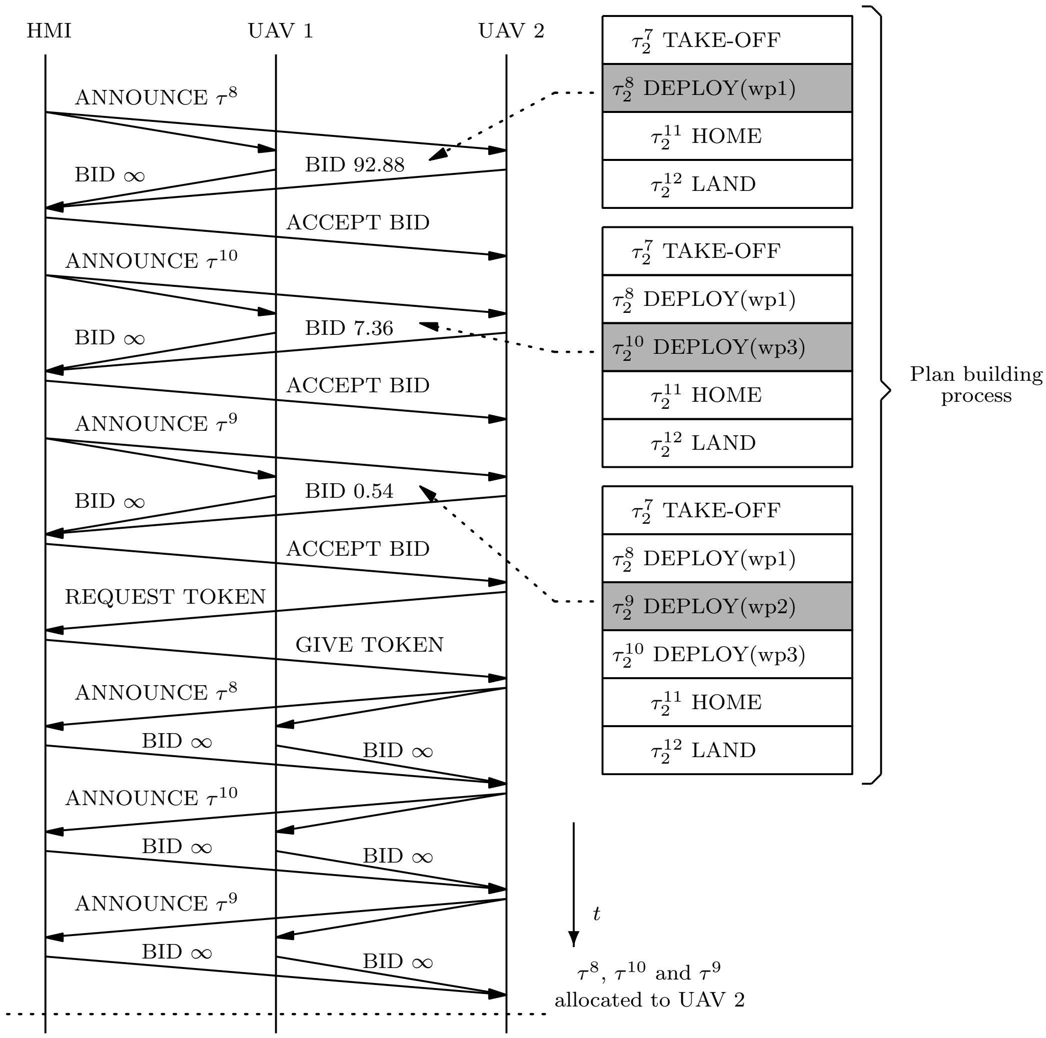


Fig. 10 CNP messages interchanged for the allocation of the sensor deployment tasks (Mission #5). The labels of the arrows representing the messages are always above them.

4. Go up again to the specified waypoint altitude.

Once decomposed, the following twelve elementary tasks were inserted in the plan, substituting tasks τ_2^8, τ_2^9 and τ_2^{10}:

- $\tau_2^8 \rightarrow \{{}^1\hat{\tau}_2^8, {}^2\hat{\tau}_2^8, {}^3\hat{\tau}_2^8, {}^4\hat{\tau}_2^8\}$ ($\hat{\lambda}^8 = \texttt{GOTO}$)
- $\tau_2^9 \rightarrow \{{}^1\hat{\tau}_2^9, {}^2\hat{\tau}_2^9, {}^3\hat{\tau}_2^9, {}^4\hat{\tau}_2^9\}$ ($\hat{\lambda}^9 = \texttt{GOTO}$)
- $\tau_2^{10} \rightarrow \{{}^1\hat{\tau}_2^{10}, {}^2\hat{\tau}_2^{10}, {}^3\hat{\tau}_2^{10}, {}^4\hat{\tau}_2^{10}\}$ ($\hat{\lambda}^{10} = \texttt{GOTO}$)

After the execution of the deployment tasks, during the landing maneuver of UAV 2, a new fire alarm was declared by one of the sensors deployed in the area between the building and the barrels. Then, in order to confirm this second fire, a take-shot

The original publication is available at www.springerlink.com in this link:
http://dx.doi.org/10.1007/s10846-010-9497-5

Table 7 Tasks executed during Mission #5 and their decomposition in elementary tasks. The values of the parameters $\hat{\Pi}^k$ corresponding to the elementary tasks with type $\hat{\lambda}^k$ = GOTO of the UAV 1 are detailed in Table 8.

τ_i^k	λ	$^{-}\Omega$	Ω^{+}	Decomposition	$\hat{\Pi}$
τ_1^1	TAKE-OFF	PRE-FLIGHT_CHECK	$\emptyset$	${}^1\hat{\tau}_1^1$ ($\hat{\lambda}^1$= TAKE-OFF)	${}^1\hat{\Pi}_1^1$
τ_1^2	TAKE-SHOT	END(τ_1^1)	$\emptyset$	${}^1\hat{\tau}_1^2$ ($\hat{\lambda}^2$= GOTO)	${}^1\hat{\Pi}_1^2$
τ_1^3	GOTO	END(τ_1^2)	$\emptyset$	${}^1\hat{\tau}_1^3$ ($\hat{\lambda}^3$= GOTO)	${}^1\hat{\Pi}_1^3$
τ_1^4	TAKE-SHOT	END(τ_1^3)	$\emptyset$	${}^1\hat{\tau}_1^4$ ($\hat{\lambda}^4$= GOTO)	${}^1\hat{\Pi}_1^4$
τ_1^5	HOME	END(τ_1^4)	$\emptyset$	${}^1\hat{\tau}_1^5$ ($\hat{\lambda}^5$= GOTO)	${}^1\hat{\Pi}_1^5$
τ_1^6	LAND	END(τ_1^5)	$\emptyset$	${}^1\hat{\tau}_1^6$ ($\hat{\lambda}^6$= LAND)	${}^1\hat{\Pi}_1^6$
τ_2^7	TAKE-OFF	PRE-FLIGHT_CHECK	$\emptyset$	${}^1\hat{\tau}_2^7$ ($\hat{\lambda}^7$= TAKE-OFF)	${}^1\hat{\Pi}_2^7$
τ_2^8	DEPLOY(wp1)	END(τ_2^7)	$\emptyset$	$\{{}^1\hat{\tau}_2^8, {}^2\hat{\tau}_2^8, {}^3\hat{\tau}_2^8, {}^4\hat{\tau}_2^8\}$ ($\hat{\lambda}^8$= GOTO)	${}^1\hat{\Pi}_2^8$
τ_2^9	DEPLOY(wp2)	END(τ_2^8)	$\emptyset$	$\{{}^1\hat{\tau}_2^9, {}^2\hat{\tau}_2^9, {}^3\hat{\tau}_2^9, {}^4\hat{\tau}_2^9\}$ ($\hat{\lambda}^9$= GOTO)	${}^1\hat{\Pi}_2^9$
τ_2^{10}	DEPLOY(wp3)	END(τ_2^9)	$\emptyset$	$\{{}^1\hat{\tau}_2^{10}, {}^2\hat{\tau}_2^{10}, {}^3\hat{\tau}_2^{10}, {}^4\hat{\tau}_2^{10}\}$ ($\hat{\lambda}^{10}$= GOTO)	${}^1\hat{\Pi}_2^{10}$
τ_2^{11}	HOME	END(τ_2^{10})	$\emptyset$	${}^1\hat{\tau}_2^{11}$ ($\hat{\lambda}^{11}$= GOTO)	${}^1\hat{\Pi}_2^{11}$
τ_2^{12}	LAND	END(τ_2^{11})	$\emptyset$	${}^1\hat{\tau}_2^{12}$ ($\hat{\lambda}^{12}$= LAND)	${}^1\hat{\Pi}_2^{12}$

Table 8 Values of the parameters $\hat{\Pi}_1^k$ corresponding to the elementary tasks with type $\hat{\lambda}_1^k$ = GOTO in Mission #5 for UAV 1. Table 4 details the meaning of each parameter π^j.

Parameters ($\hat{\Pi}_i^k$)	${}^1\hat{\Pi}_1^2$	${}^1\hat{\Pi}_1^3$	${}^1\hat{\Pi}_1^4$	${}^1\hat{\Pi}_1^5$
π^1	251688.49	251665.36	251689.67	251674.17
π^2	4121282.87	4121282.99	4121283.58	4121244.74
π^3	75.0	70.0	75.0	70.4
π^4	1.0	1.0	1.0	1.0
π^5	1	1	1	1
π^6	90.0	90.0	90.0	0.0
π^7	0	0	0	0

task (τ^2) was specified: take images from the west side of the fire at an altitude of 75 meters. A negotiation process started, and the task was allocated to UAV 1 (τ_1^2), which was equipped with an infrared camera (UAV 2 had no cameras on-board and its bids were infinite).

Task τ_1^2 was processed by the plan refining toolbox (see Fig. 2) in order to compute the waypoint that fulfilled the above constraints and allowed to have the fire in the center of the field of view of the on-board camera. Once the fire was confirmed, the platform operator commanded a fire truck equipped with a remotely controlled extinguisher (water cannon) to extinguish it. Before activating the extinguisher, UAV 1 was commanded to a safe location (task τ_1^3). After the operation with the water cannon, the user commanded again a take-shot task τ_1^4 for UAV 1 in order to confirm that the fire was extinguished. After the confirmation, the UAV returned home and landed (tasks τ_1^5 and τ_1^6).

Table 7 summarizes all the tasks described above, along with their decomposition into elementary tasks. Moreover, from the elementary tasks allocated to UAV1, those with type $\hat{\lambda}^k$ = GOTO are detailed in Table 8. Table 4 details the meaning of each parameter π^j. The values of the π^1 and π^2 parameters shown in Table 8 are represented in Fig. 11 as small red squares.

The original publication is available at www.springerlink.com in this link:
http://dx.doi.org/10.1007/s10846-010-9497-5

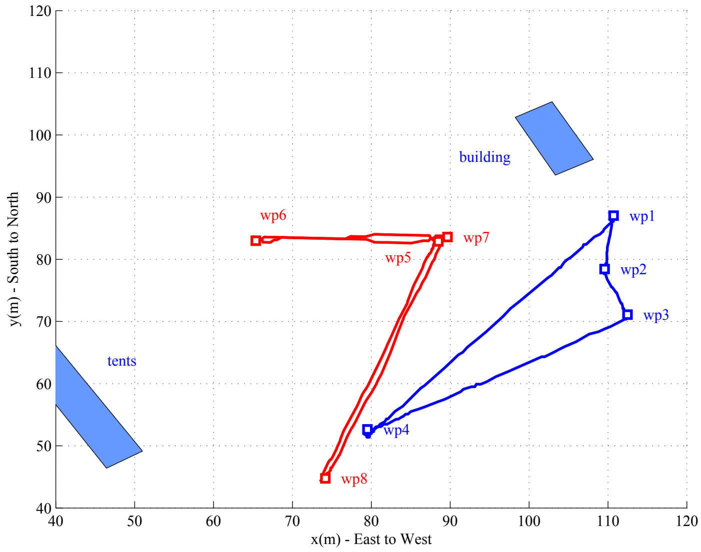


Fig. 11 Paths followed by the two helicopters during the node deployment and fire monitoring mission (Mission #5).

Figure 11 shows the paths followed by the two helicopters (red and blue for the UAVs 1 and 2 respectively). The small squares represent the waypoints corresponding to the elementary GOTO tasks:

- UAV 1 (red line):
  - `wp5, wp7`: locations computed to monitor the fire.
  - `wp6`: safe waypoint to wait for the water cannon operation to be over.
  - `wp8`: UAV 1 home.
- UAV 2 (blue line):
  - `wp1, wp2, wp3`: deployment locations.
  - `wp4`: UAV 2 home.

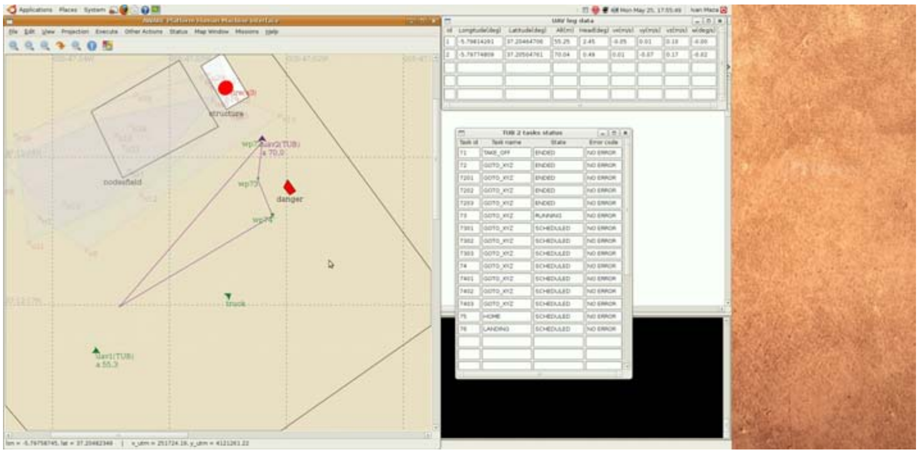
Finally, Figure 12 shows different screenshots of the HMI application taken during the execution of the mission. Figure 12(a) shows UAV 2 in operation during the execution of the three sensor deployment tasks transformed into twelve elementary goto tasks (see "TUB2 tasks status" window). On the other hand, in Fig. 12(a) UAV 1 is monitoring the fire detected by the sensors deployed: a window shows the images captured by the infrared camera on-board with a red overlay corresponding to the fire detected.

5.2 Multi-UAV Surveillance

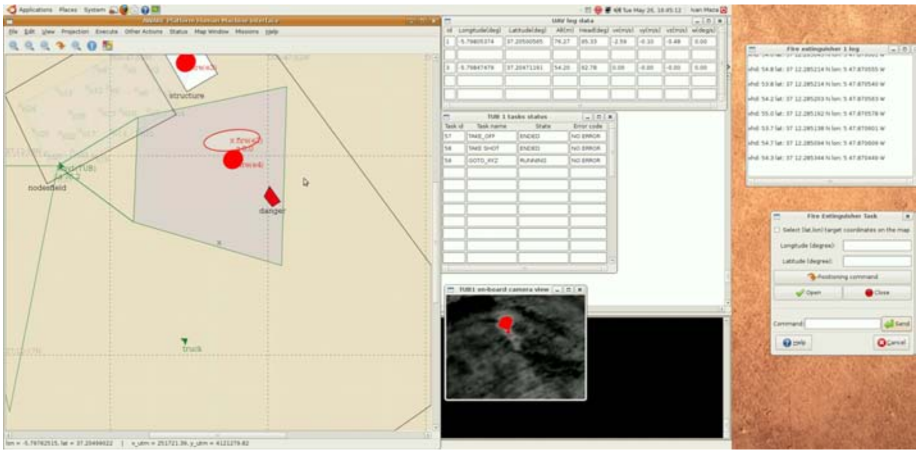
In this mission (identified as #7 in Table 5), the objective was to find objects of interest in a given area. In our case, the objects of interest were fuel barrels located around a building where a fire alarm had been declared.

The propagation of the fire could reach those barrels and make more difficult the extinguishing task. Then, the platform user specified a surveillance task τ^1 to localize

The original publication is available at www.springerlink.com in this link:
http://dx.doi.org/10.1007/s10846-010-9497-5



(a) A fire had been declared in the building and three sensors were deployed to detect its potential propagation to the fuel tanks close to the building



(b) The fire propagation was detected with the sensors previously deployed and a second UAV equipped with an infrared camera took-off to confirm it and to provide estimations of the evolution of the fire

Fig. 12 Screenshots of the platform human machine interface during the execution of Mission #5: sensor deployment and fire monitoring. The screenshot on the top shows UAV 2 in operation during the execution of the three sensor deployment tasks transformed into twelve elementary goto tasks (see "TUB2 tasks status" window). On the other hand, in the screenshot below UAV 1 is monitoring the fire detected by the sensors deployed: a window shows the images captured by the infrared camera on-board with an red overlay corresponding to the fire detected.

the barrels and display them on the map of the HMI. In this mission, two UAVs were available and ready on the landing pads. Both were equipped with a fixed visual camera aligned with the fuselage of the helicopter and pointing downwards 90°.

The values for the parameters of the surveillance task τ^1 are shown in Table 9 and the meaning of each parameter is explained in Table 10. Basically, the user specifies the vertices of the area to be covered, the altitude for the flight, the speed for the UAVs

The original publication is available at www.springerlink.com in this link:
http://dx.doi.org/10.1007/s10846-010-9497-5

Table 9 Values for the tasks parameters (Π^k). The meaning of each parameter π^j is explained in Table 10.

Parameters (Π^k)	Π^1
π^1	251685.14 4121220.82 251663.65 4121237.68 251655.84 4121263.99 251666.46 4121282.96 251694.82 4121288.55 251721.20 4121288.71 251731.66 4121269.75 251728.22 4121246.46 251706.54 4121231.88
π^2	72.0
π^3	1.0
π^4	100.0

Table 10 Parameters of a task with type λ = SURV.

Parameters (Π^k)	Description
π^1 (Polygon)	The set of vertices defining the polygon of the area to be covered by the UAV
π^2(Altitude)	Altitude (m) for the flight (ellipsoid-based datum WGS84)
π^3(Speed)	Specified speed (m/s) for the flight
π^4(Overlapping)	Overlapping in percentage between consecutive rows of the zigzag pattern

and the desired overlapping between images taken in consecutive rows of the searching zigzag pattern.

In the distributed negotiation for the surveillance task, the different bids are used by the auctioneer to compute the relative capabilities of the available UAVs for the area partition process. The idea is to divide the whole area specified by the user among the available UAVs taking into account their relative capabilities following the approach described in [10, 14].

Then, once the surveillance task was announced by the HMI application, the two available UAVs started with the negotiation process bidding with their particular capabilities for the execution. Each bid was computed by the plan refining toolbox module taking into account the specified altitude and the parameters of the on-board cameras as

$$b_i = w_i P_{d_i}, \tag{1}$$

where P_{d_i} was the probability of detection for the object of interest and w_i was the sensing width of the on-board camera. This width can be computed from the camera intrinsic matrix $\mathbf{C}$ given by

$$\mathbf{C} = \begin{bmatrix} \alpha_u & \gamma & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

with (u_0, v_0) the coordinates of the principal point, α_u and α_v the scale factors in image u and v axes, and γ the parameter describing the skewness of the image axes.

The original publication is available at www.springerlink.com in this link:
http://dx.doi.org/10.1007/s10846-010-9497-5

Table 11 Parameters of the cameras on-board during the surveillance mission.

Camera parameters	UAV 1	UAV 2
width (pixels)	384	384
height (pixels)	288	288
u_0	199.9948	179.9591
v_0	116.6379	112.6779
α_u	551.3304	494.4553
α_v	549.3181	492.6934
γ	0.0026	0.0017

Table 12 Values for the bids and resulting relative capabilities in percentage.

	w_i	P_{d_i}	b_i	Relative capability
UAV 1	5.17	0.901	4.65	50.12 %
UAV 2	5.76	0.803	4.63	49.88 %

In particular, Table 11 shows the values of these parameters for the cameras used in this mission.

The particular values computed in Mission #7 for the bids and the relative capabilities are shown in Table 12. In the last column of the table, the values for the relative capabilities computed by the auctioneer determine the percentage of the full area that was assigned to each UAV.

Once each UAV received from the auctioneer the relative capabilities, it could compute the whole partition and its assigned sub-area. The plan refining module of each UAV also computed the list of goto tasks required to cover the allocated sub-area based on the sensorial capabilities of each UAV and the flight altitude. Then, Figure 13 shows the location of the waypoints computed by each UAV, once the partition of the full area specified in τ^1 was done.

It should be mentioned that given the sensing widths w_i shown in Table 12, the waypoints were computed taking into account a 100% overlapping specified between consecutive rows. But, it can be seen that in the frontier between sub-areas the distance between rows of different UAVs is larger. This difference comes from the 0% overlapping that is forced between sub-areas in order to increase the safety conditions during the flights.

Finally, Figure 14 shows three screenshots captured from the HMI application during the execution of the mission. On the right of each screen, there are two windows with the images received from the cameras on-board. In Fig. 14(a), two barrels are in the field of view of the UAV2 camera, allowing to estimate their locations. Later, in Fig. 14(b), the computed estimations of the locations of the barrels are shown on the map as red dots.

6 Conclusions

This paper has presented a distributed decision-making architecture suitable for multi-UAV coordination in very challenging scenarios such as disaster management or civil security. The experiments with real UAVs presented in the paper have shown that the

The original publication is available at www.springerlink.com in this link:
http://dx.doi.org/10.1007/s10846-010-9497-5

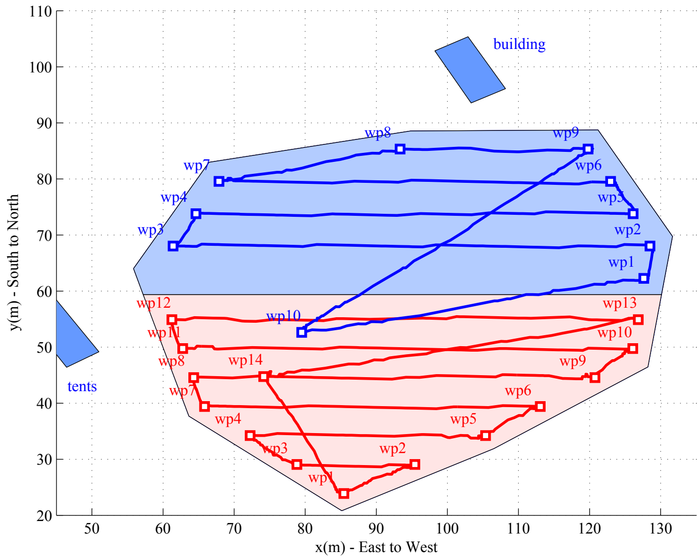


Fig. 13 Paths followed by the two helicopters during the multi-UAV surveillance mission (Mission #7).

developed architecture allows to cover a good spectrum of missions: surveillance, sensor deployment, fire confirmation and extinguishing.

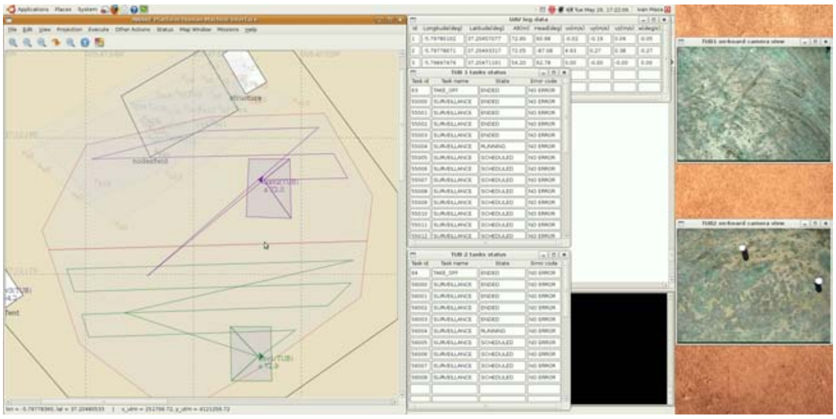
One of the key features of the architecture was the easy integration process of autonomous vehicles from different manufacturers and research groups. This characteristic made possible the integration of different types of UAVs during the AWARE Project with low development efforts.

It is worth to mention that the HMI application was shut down and restarted during the execution of several missions and the platform performance was not affected at all due to the distributed nature of the decision making system. In addition, the design did not pose significant restrictions to the communication layer, so the coordination among the UAVs using a distributed approach was possible at a reasonable communication cost. Thus, the authors consider that the proposed approach is a good balance between robustness, optimal decision making and communication resources.

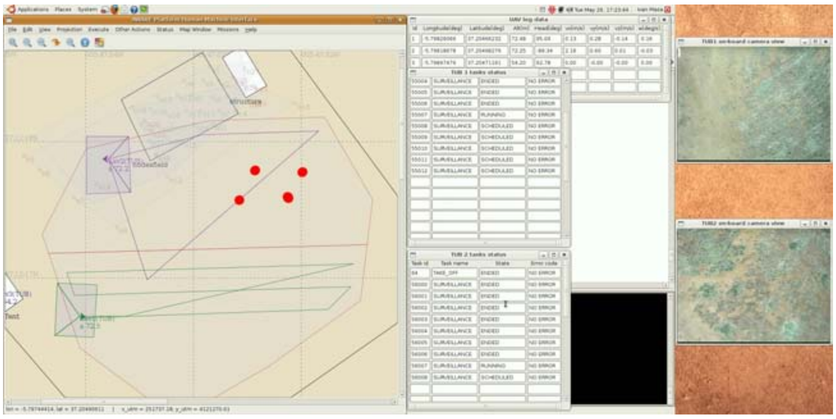
Regarding future developments, the practical application of a team of aerial vehicles will require the integration with piloted aerial vehicles. In fact, this is clear in disaster management and civil security applications. In the real scenario, piloted airborne means, i.e. airplanes and helicopters, are used today in disaster management activities. Then, the coordination of these aerial means with the unmanned aerial vehicles is a must, and the architecture presented in this paper should be extended for the integration with the conventional aircrafts.

Acknowledgements The authors thank the support of the partners of the AWARE Project; without their contributions and help, the AWARE General Experiments would have never been possible. The authors would also like to specially acknowledge Konstantin Kondak and Markus Bernard from Technische Universität Berlin (TUB), and Emmanuel Previnaire from Flying Cam (FC). They provided the UAVs that allowed to test and validate the decisional

The original publication is available at www.springerlink.com in this link:
http://dx.doi.org/10.1007/s10846-010-9497-5



(a) Two barrels in the images from the camera on-board UAV 2 on the right



(b) The estimation of the position of the barrels was computed and represented by red dots

Fig. 14 Screenshots of the platform Human Machine Interface during the execution of surveillance mission (Mission #7).

architecture presented in this paper, and their excellent work during the experiments was crucial for the success in the different missions.

References

1. Botelho, S.C., Alami, R.: M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement. In: Proceedings of the IEEE International Conference on Robotics and Automation, vol. 2, pp. 1234–1239. Detroit, USA (1999)
2. Caballero, F., Merino, L., Gil, P., Maza, I., Ollero, A.: A probabilistic framework for entire WSN localization using a mobile robot. Robotics and Autonomous Systems **56**(10), 798–806 (2008). DOI http://dx.doi.org/10.1016/j.robot.2008.06.003
3. Caballero, F., Merino, L., Maza, I., Ollero, A.: A particle filtering method for wireless sensor network localization with an aerial robot beacon. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 596–601. Pasadena, California, USA (2008). DOI http://dx.doi.org/10.1109/ROBOT.2008.4543271

The original publication is available at www.springerlink.com in this link:
http://dx.doi.org/10.1007/s10846-010-9497-5

4. Capitan, J., Merino, L., Caballero, F., Ollero, A.: Delayed-State Information Filter for Cooperative Decentralized Tracking. In: Proceedings of the International Conference on Robotics and Automation, pp. 3865–3870 (2009)
5. Dias, M.B., Stenz, A.: Opportunistic optimization for market-based multirobot control. In: Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2714–2720. Lausanne, Switzerland (2002)
6. Gerkey, B., Matarić, M.: Sold!: Auction methods for multi-robot coordination. IEEE Transactions on Robotics and Automation **18**(5), 758–768 (2002)
7. Gerkey, B., Matarić, M.: A formal analysis and taxonomy of task allocation in multi-robot systems. International Journal of Robotics Research **23**(9), 939–954 (2004)
8. Gerkey, B.P., Matarić, M.J.: Multi-robot task allocation: Analyzing the complexity and optimality of key architectures. In: Proceedings of the IEEE International Conference on Robotics and Automation, vol. 3, pp. 3862–3868. Taipei, Taiwan (2003)
9. Heredia, G., Caballero, F., Maza, I., Merino, L., Viguria, A., Ollero, A.: Multi-unmanned aerial vehicle (UAV) cooperative fault detection employing differential global positioning (DGPS), inertial and vision sensors. Sensors **9**(9), 7566–7579 (2009). DOI 10.3390/s90907566
10. Hert, S., Lumelsky, V.: Polygon area decomposition for multiple-robot workspace division. International Journal of Computational Geometry and Applications **8**(4), 437–466 (2001)
11. Konolige, K., Fox, D., Limketkai, B., Ko, J., Stewart, B.: Map merging for distributed robot navigation. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 212–217 (2003)
12. Kontitsis, M., Garcia, R., Valavanis, K.: Design, implementation and testing of a vision system for small unmanned vertical take off and landing vehicles with strict payload limitations. Journal of Intelligent and Robotic Systems **44**(2), 139–159 (2005)
13. Maza, I., Kondak, K., Bernard, M., Ollero, A.: Multi-UAV cooperation and control for load transportation and deployment. Journal of Intelligent and Robotic Systems **57**(1–4), 417–449 (2010). DOI 10.1007/s10846-009-9352-8
14. Maza, I., Ollero, A.: Distributed Autonomous Robotic Systems 6, *Distributed Autonomous Robotic Systems*, vol. 6, chap. Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms, pp. 221–230. Springer Verlag (2007)
15. Merino, L., Caballero, F., de Dios, J.M., Ferruz, J., Ollero, A.: A cooperative perception system for multiple UAVs: Application to automatic detection of forest fires. Journal of Field Robotics **23**(3–4), 165–184 (2006)
16. Ollero, A., Lacroix, S., Merino, L., Gancet, J., Wiklund, J., Remuss, V., Veiga, I., Gutierrez, L.G., Viegas, D.X., A.Gonzalez, M., Mallet, A., Alami, R., Chatila, R., Hommel, G., Colmenero, F.J., Arrue, B., Ferruz, J., Martinez, J.R., Caballero, F.: Multiple eyes in the sky: Architecture and perception issues in the COMETS unmanned air vehicles project. IEEE Robotics and Automation Magazine **12**(2), 46–57 (2005)
17. Ollero, A., Maza, I. (eds.): Multiple heterogeneous unmanned aerial vehicles. Springer Tracts on Advanced Robotics. Springer-Verlag (2007)
18. Parker, L.: ALLIANCE: An architecture for fault-tolerant multi-robot cooperation. IEEE Transactions on Robotics and Automation **14**(2), 220–240 (1998)
19. Parker, L.: Distributed intelligence: Overview of the field and its application in multi-robot systems. Journal of Physical Agents **2**(1), 5–14 (2008)
20. Smith, G.: The Contract Net Protocol: High-level communication and control in a distributed problem solver. IEEE Transactions on Computers **29**(12), 1104–1113 (1980)
21. Valavanis, K. (ed.): Advances in Unmanned Aerial Vehicles: State of the Art and the Road to Autonomy. Intelligent Systems, Control and Automation: Science and Engineering. Springer-Verlag (2007)
22. Viguria, A., Maza, I., Ollero, A.: SET: An algorithm for distributed multirobot task allocation with dynamic negotiation based on task subsets. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 3339–3344. Rome, Italy (2007). DOI 10.1109/ROBOT.2007.363988
23. Viguria, A., Maza, I., Ollero, A.: S+T: An algorithm for distributed multirobot task allocation based on services for improving robot cooperation. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 3163–3168. Pasadena, California, USA (2008). DOI 10.1109/ROBOT.2008.4543692
24. Viguria, A., Maza, I., Ollero, A.: Distributed service-based cooperation in aerial/ground robot teams applied to fire detection and extinguishing missions. Advanced Robotics **24**(1–2), 1–23 (2010)

The original publication is available at www.springerlink.com in this link:
`http://dx.doi.org/10.1007/s10846-010-9497-5`