

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Interoperability in a Heterogeneous Team of Search and Rescue Robots

Daniel Serrano López, German Moreno,
Jose Cordero, Jose Sanchez, Shashank Govindaraj,
Mario Monteiro Marques, Victor Lobo,
Stefano Fioravanti, Alberto Grati, Konrad Rudin,
Massimo Tosa, Anibal Matos, Andre Dias,
Alfredo Martins, Janusz Bedkowski, Haris Balta and
Geert De Cubber

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/intechopen.69493>

Abstract

Search and rescue missions are complex operations. A disaster scenario is generally unstructured, time-varying and unpredictable. This poses several challenges for the successful deployment of unmanned technology. The variety of operational scenarios and tasks lead to the need for multiple robots of different types, domains and sizes. A priori planning of the optimal set of assets to be deployed and the definition of their mission objectives are generally not feasible as information only becomes available during mission. The ICARUS project responds to this challenge by developing a heterogeneous team composed by different and complementary robots, dynamically cooperating as an interoperable team. This chapter describes our approach to multi-robot interoperability, understood as the ability of multiple robots to operate together, in synergy, enabling multiple teams to share data, intelligence and resources, which is the ultimate objective of ICARUS project. It also includes the analysis of the relevant standardization initiatives in multi-robot multi-domain systems, our implementation of an interoperability framework and several examples of multi-robot cooperation of the ICARUS robots in realistic search and rescue missions.

Keywords: interoperability, multi-robot collaboration

1. Introduction

There are nowadays many different types of unmanned systems being used in different domains and, certainly, this number will increase significantly in the upcoming years. In general, large scale systems aiming at solving all problems with one single type of platform have proven to be expensive and not flexible enough. Heterogeneous teams, composed by unmanned air, ground, surface, and underwater systems (UxS), of different types and sizes, offer the possibility to exploit the best features of each kind and combine them to obtain compound capabilities, which have demonstrated to be more cost-efficient and adaptable to new scenarios. Recent research efforts have focused on developing the autonomy of the team by increasing the interactions between these systems, making them aware of each other, executing tasks that require cooperation, and finally implementing flock or swarm coordinated behaviours.

The ICARUS project involves a team of assistive unmanned air, ground and sea vehicles for search and rescue operations. In order to effectively support the on-site person responsible for the operations, these systems must be able to collaborate as a seamlessly integrated team, coordinated from the ICARUS Robot Command and Control station (RC2) in the field.

A heterogeneous fleet is the one composed by elements of different kinds such as the ICARUS team, including up to ten different vehicles (long-endurance fixed-wing, outdoors multi-rotor, indoors multi-rotor, large UGV, small UGV, Teodor UGV, U-ranger USV, ROAZ USV, MARES AUV and several rescue capsules). Each robot has been developed by a different provider or partner, using its own design, framework and middleware. Thus, a strong effort had to be devoted to their integration as a team and this is the work described in this Chapter. Although many standards have been proposed by the community, most of the field robotic systems have their own command and reporting protocols, and consequently require their own ground control stations. This profusion of protocols makes the cooperation between systems difficult. The lack of unified standards poses an unnecessary burden on the operation and maintenance of multi-vehicle systems. The work described in this chapter aims at contributing to the harmonization of the multiple standardization initiatives for the coordination of heterogeneous teams.

The ultimate objective of the ICARUS project is to achieve robot interoperability, which can be understood as the ability of robots to operate in synergy to the execution of assigned missions. Interoperability enables diverse teams to work together, sharing data, intelligence and resources.

2. Approach to interoperability

Interoperability is the key that acts as the glue among the different units within the team, enabling efficient multi-robot cooperation. Seamless and non-ambiguous interaction between different robots of any provider and domain demands a common, well-defined interface.

ICARUS proposes the adaptation of all the vehicles to a single standard external interface as a method to ensure interoperability. Each robot development team is free to use their own tools inside their systems as long as the interaction with the rest of the team follows a set of definitions and rules referred to as the interoperability standard. This follows the façade pattern [1] very frequently used in software engineering. It essentially hides the complexities of the implementation and provides the outer components with a simpler interface. It is typically deployed as software library implementing a wrapper or adapter template. On one side, this library implements the interoperability standard interface and, on the other side, it provides a set of classes and functions (an API) for its integration with the specific middleware or software provided by each platform.

This approach may initially seem to reduce the level of integration among the agents if we compare it against natively sharing an internal protocol in all systems, but it promotes the maximum decoupling between the custom implementations, with its particularities, and the definition of the common interface. In the long term, this has shown to improve the seamless integration of the maximum number of systems and domains at a lower cost. The integration of new platforms into the team has literally been done in a matter of few hours during the project, provided that on-board hardware resources and communications are made available by the robot provider.

Therefore, the ultimate goal of the work on the heterogeneous team is to consolidate a common command, control and payload interface to be agreed and adopted by all robotics platforms and control ground stations (CGS) involved in an ICARUS operation. This approach provides a common framework for the development of collaborative unmanned assets, minimizing the integration time and costs by avoiding ad-hoc implementations.

There are other advantages in using interoperability standards. The use of a widely accepted interface helps to easily integrate new technologies with minor modifications to the existing systems. This facilitates the insertion of new technology for their operational use in the field, as end-users rely on proven technology and the preliminary validation will focus only on de-risking the new developments. Another advantage of the use of standards is that it will facilitate the backwards and forwards compatibility between existing and future vehicles and CGS provided by different providers. This can benefit companies to maximize the revenue from a specific product.

Our strategy in terms of interoperability is to build upon existing body of work in the field, avoiding duplicating and re-inventing proven technology. During the initial steps of the work, the most relevant multi-domain interoperability protocols for unmanned systems were identified and evaluated against the ICARUS end-user requirements and foreseen scenarios. During this phase, several collaborations with other European [2] and NATO [3] initiatives, together with the organization of workshops involving end-users and stakeholders, were extremely relevant to gather good quality information on the state of the art in the field.

2.1. Ontology definition

One of the challenges in multi-robot multi-domain interface standardization is to be able to embrace all type of systems, independently of their domain, particularities (i.e. size, operational

modes, etc.) or constraints (i.e. computational resources, communication bandwidth, etc.). Therefore, in order to methodically evaluate the existing initiatives, an analysis of the ICARUS robot specific interfaces control (ICD) and functional specifications (FSD) was performed to generate what we referred to as the project interoperability needs. Any information that was domain- or platform- specific was removed from the analysis to ensure the level of abstraction required to ensure standardization. Likewise, the needs were further developed through an analysis of other potential vehicles that could be integrated into the system in the future.

This set of needs has been formalized as an ontology. An ontology is ‘an explicit, formal specification of a shared conceptualization’ [4]. We use it to describe the set of concepts required to coordinate a multi-robot search and rescue operation. This includes concepts, at different levels from robots to systems, capabilities and sensors, and their relationships and assumptions. There have been previous and parallel efforts in this field. Namely, the IEEE Robotics and Automation Society (IEEE-RAS) created a working group named Ontologies for Robotics and Automation that aims at the definition of a core ontology for robotics and automation [5]. The work performed in ICARUS has a strict focus on heterogeneous multi-robot operations in search and rescue, and as such, it proposes an application-specific ontology, addressing tasks and platforms involved in search and rescue missions.

This analysis resulted in a description of the set of multi-domain concepts and relationships or messages commonly found in unmanned systems. **Table 1** summarizes the key categories and provides some example of interactions between systems.

The complete ontology was used in a gap-analysis for the evaluation of the existing standards as described in Section 3.

2.2. Interoperability levels

A key concept that enables interoperability among the largest number possible of unmanned systems is the levels of interoperability (LoI). This concept is introduced by STANAG 4586

Category	Description and examples
Transport	Inter-process communication such as send, receive, broadcast, etc.
Commands	Generic accessories such as set, get, etc. for any standard concept
Management	Heartbeat, system status, clock synchronization, alarms, etc.
Telemetry	Pose and velocity reports in appropriate system coordinates, etc.
Telecontrol	Teleoperation, waypoint and mission management, etc.
Perception	Imagery, ranging, audio, etc.
Manipulation	Joint and end-effector control of robotics arms
Mapping	Maps, digital elevation models, point clouds
S & Rintelligence	Sectors, disaster alerts, humanitarian information

Table 1. Examples of the ICARUS ontology concepts organized by categories.

[6] and has been adopted in ICARUS and adapted for the purpose of our project. LoI defines the different degrees of compliance with the standard interface. It proposes a mechanism to account for a large variety of approaches and levels at which different systems can be integrated, accounting therefore for more integration strategies and combinations.

STANAG 4586 defines LoI as ‘the platform, subsystem or sensor ability to be interoperable for basic types of functions related to unmanned systems’. These levels show different degrees of control that a user has over the vehicle, payload or both. However, these definitions have been adapted to our project as follows.

Therefore, the levels of interoperability in ICARUS are defined as shown in **Table 2**.

The levels of interoperability for each of the ICARUS systems are as shown in **Table 3**.

2.3. Adjustable automation

ICARUS is, by definition, a human-centred designed system. One of the most critical end-user requirements is to ensure that a member of the search and rescue team in the field always supervises the robot operations to ensure safety and effectiveness. ICARUS robotics asset can generally be remotely controlled. Most of these systems also provide on-board autonomy modules that allow the operator to plan a mission to be autonomously executed by the system. This should presumably help reducing the workload of the operator. However, in a realistic scenario, unexpected events are highly likely to occur and the intervention from the operator, such as manually overriding the mission execution, is often required. This increases the cognitive workload of the operator leading to stress and potential mistakes, which are even more critical in the context of multi-robot operations.

Adjustable automation (AA) is the ability of a robot to behave autonomously and dynamically change its level of independence, intelligence and controllability to adapt to different tasks and scenarios [7]. AA presents advantages when dealing with communication delays, human workload and safety [8]. Having systems that can dynamically reduce or increase the level of automation running on board provides a more flexible and reliable system.

In ICARUS, AA is achieved by supporting multiple levels of automation in the robots, e.g. fully autonomous, guided by the operator, and fully controlled by the operator. The C2I also

Level of interoperability	
LoI 1	Indirect receipt/transmission of telemetry, control and payload data: the UxV data are received from (or sent to) another source (another CGS, web-server, etc.)
LoI 2	Direct receipt/transmission of UxV telemetry and payload data, but without control authority over it
LoI 3	Direct control and monitoring over the UxV without launch and recovery. A dedicated control station keeps control for the safety critical operations of the platform (i.e. take-off and landing, deployment and recovery, etc) and hand it over to the CGS once ready for mission
LoI 4	Highest level of interoperability. The CGS has full control of the UxV

Table 2. ICARUS definition of the levels of interoperability.

Level of automation per robot					
	LoI 1	LoI 2	LoI 3	LoI 4	Notes
Long-endurance fixed-wing			X		Take-off and landing procedures for the UAVs are handled by the proprietary control stations. The system is handed over to the ICARUS C2I once in air
Outdoors multi-rotor			X		
Indoors multi-rotor			X		
Large UGV				X	U-ranger is a highly equipped and extremely fast USV. Integration is done through its proprietary CGS for safety purposes
Small UGV				X	
U-ranger USV	X				
ROAZ USV			X		ROAZ USV is primarily operated from a proprietary CGS. When ICARUS mission starts, control is handed over to the ICARUS C2I
Rescue capsule				X	

Table 3. LoI for each of the robots in ICARUS.

supports adjustable automation by automatically changing its display and control functions based on the relevance of the information, the current situation the robots encounter, and user preferences.

The level of automation of a robot is related to the degree of intervention of the human operator and other robots in the decision process. However, the fact that a robot is autonomous does not imply that it has to make all its decisions by itself. Different levels of automation and classifications have been described in the literature [9]. Specifically, Lacroix et al. [10] defines five levels of automation according to the robot responsibilities towards a fleet of robots (tasks allocation, mission coordination, etc), which are mostly relevant for tightly coupled coordination. In ICARUS, the levels of automation are understood in terms of tasks execution and are reduced to essentially three modes, as shown in **Table 4**.

An ICARUS platform can seamlessly carry out a given task at different automation levels, depending on the robot operator choice, the mission plan priorities, workload and constraints of the mission and platform. As mentioned before, the concept of adjustable autonomy implies the ability to adapt and dynamically change between these levels of autonomy depending on situational changes. Some examples of adjustable autonomy within the context of ICARUS are:

- A UAV may provide fully-autonomous navigation in nominal conditions, but may fall back to semi-autonomous navigation in the presence of victims detected on the sensor stream.
- The RC2 operator may have initially designed the mission to manually operate the outdoor multi-rotor to inspect a building, but operation enters a highly complex area and he/she decides to enable semi-autonomous mode to ensure all corners are correctly surveyed.

Level of automation	
Level 1	Teleoperation. No automation on-board the robot. The robot is directly controlled by the operator
Level 2	Semi-autonomous. Execution capabilities. The robot is able to manage partially ordered sequences of elementary tasks, and to return execution status of the tasks. An operator is supervising the mission from the RC2
Level 3	Fully-autonomous. Deliberative capabilities. Complex task requests are managed (tasks planning and scheduling)

Table 4. ICARUS definition of the level of automation.

Most ICARUS platforms provide all three operation modes. However, there are specific constraints in some platforms due to their size or domain. Namely, the large UGV is usually remotely operated or waypoint guided. Such a large system should not be tasked with pre-defined missions. On the other hand, the U-ranger is such a fast maritime system that the operator should rely on the on-board autonomy, which is equipped with collision avoidance functionality. Obstacles at sea are difficult to see by the operator and therefore, this system is better commanded at full automation. **Table 5** illustrates the automation levels available in each of the ICARUS platforms.

2.4. Multi-robot cooperation

An effective heterogeneous team management requires the capability to do reasoning about the mission goals in order to provide a task-to-robot decomposition. This task allocation must take into account the current capabilities and constraints of each asset in the team. Different strategies for the cooperation are feasible and, therefore, different requirements may be placed on the interface in order to implement these strategies. A heterogeneous team usually contains a set of vehicles with diverse capabilities that can therefore play different roles in the mission. Concepts such as roles, responsibilities, modes of operations and tasks may be part of the standard interface that supports the fleet interoperability.

The platforms involved in ICARUS have been carefully selected to help each other. In other words, they play complementary roles. Several ICARUS platforms grouped together form a team. Each vehicle has been designed to provide a set of specific functionalities but they can address more complex missions by supporting each other.

Level of automation per robot								
	Long-endurance fixed-wing	Outdoor multi-rotor	Indoor multi-rotor	Large UGV	Small UGV	U-Ranger USV	ROAZ USV	Rescue Capsule
Level 1								
Level 2								
Level 3								

Table 5. Level of automation for each of the robots in ICARUS.

Different strategies for the coordination are feasible. In the case of ICARUS, a strong end-users need is that any planning decision must be authorized by the on-site operations coordinator. Therefore, according to a traditional classification of multi-robot systems based on the coordination strategy [11], ICARUS follows a supervised, weakly-coordinated, centralized approach where the cooperation and interaction between robots is negotiated during mission planning. The planning, coordination, and therefore the ultimate responsibility fall on the ICARUS team operator and occur at the C2I. Therefore, this coordination approach relaxes to a certain extent the need to have multi-robot related concepts in the interoperable interface. The C2I encapsulates this functionality and can interact with each asset individually. However, a standard for coordinated multi-robot operations remains extremely relevant and was taken into account in the analysis described in the next section.

Some of key concepts to be unambiguously defined as the basis for efficient mission planning are goal, role and task. A mission goal refers to the overall objective that the fleet must accomplish, for instance, the assessment of a disaster area. The mission planner is responsible for coordinating the fleet and allocating specific roles to each robot. A role defines the robot's behaviour and its interactions with other members of the fleet or with humans. A task is the basic unit describing the actions requested from a robot. Typically, the role defines which tasks a robot should and should not execute. A robot is defined by the type of robot and its capabilities. For instance, the long-endurance is a fixed-wing aerial platform with surveillance, mapping, victim detection and communications relay capabilities. These characteristics define the set of tasks that it is able to perform, and therefore the roles it can take. A mission plan is therefore built upon the concepts of roles, tasks and responsibilities.

ICARUS planning flow mirrors the concept of operation of international search and rescue teams. **Table 6** illustrates how goals to roles and task decomposition occur on the field.

One of the core services required from all the platforms is the dynamic discovery of features. It allows robots to advertise their capabilities over the network, enabling dynamic planning and

System	Responsibilities
C2I	Sectorization + mission goals definition
	SAR team allocation to sectors
	Robot(s) allocation to teams
	Teams monitoring and control
RC2	Operations scheduling
	Roles allocation
	Robot task planning
	Robots monitoring and control
Robot	Task plan execution
	Progress and status report

Table 6. ICARUS goals to roles and tasks decomposition.

supervision from the C2I based on the current state of the team. A robot may take different roles during a mission depending on the responsibilities that the C2I allocates to it. The allocation of mission goals to predefined roles, the decomposition of these roles into tasks, and the configuration of these tasks for a specific robot model are the responsibility of the mission planner. Some predefined profiles are available to facilitate this task. Whereas roles influence the robots' behaviour, tasks influence the actions that robots perform. They are defined as a set of actions. Each task could be decomposed into subtasks. This subdivision could continue iteratively until a primitive task is reached. **Table 7** shows some examples of the roles defined in the ICARUS concept of operations.

Roles	Description	Modes
Scout	Provides a quick assessment of an unexplored area or route	Overview of an entire disaster zone (fixed-wing UAS). Traversability/best route exploration (UAS)
Surveyor	Scan in detail an area or building to support a thorough assessment and inspection (structural integrity, victims, hazards, etc.)	2D/3D geo-referenced map of the entire disaster zone as basis for sectorization (fixed-wing UAS). High-resolution 2D/3D geo-referenced map of a sector (fixed-wing UAS at higher altitude, rotor-craft at lower altitude) or a structure (rotor-craft). Building indoor inspections (small rotor-craft and small ground vehicle)
Observer	Steady target observation and assessment, including victims and structures	Steady hover over a target (rotor-craft), including harsh weather conditions. Victim medical assessment outdoors (rotor-craft and USV) and indoors (small rotor-craft and ground vehicles)
Searcher	Victims search	Outdoors human detection on IR (UAS and USV), indoors (small rotor-craft and UGV)
Rescuer	Support to victim rescue	Helps victim to escape from hazard areas (large ground vehicle) or support human rescuers
Deliverer	Safety kit delivery. Robot delivery	Delivery of a survival kit to a victim, aerial (rotor-craft) or terrestrial (UGV)
Cruiser	Travel to a destination	All platforms when transiting to a new location where another role is enabled. The larger platforms may also act as a carrier of tools, debris and smaller robotics assets

Table 7. Examples of ICARUS roles.

3. Analysis of existing standardization initiatives

ISO defines a standard as a set of 'requirements, specifications, guidelines or characteristics that can be used consistently to ensure that materials, products, processes and services are fit for their purpose' [12]. In the context of interoperability, a standard shall unambiguously define data types, message and rules to implement the protocol. The analysis of the existing standardization initiatives shows that there exist several predominant initiatives for interoperability of unmanned systems [3]. However, harmonization among them is not yet a fact.

In the context of this analysis, we divided the different initiatives in two different groups:

- Fully operational standards and
- Partially operational resources.

The first group focuses on systems interoperability, providing a common communication framework between different agents. They provide all the basic functionality required for a multiplatform system. The second group includes initiatives that are, either very popular on specific fields, or are designed specifically for some particular tasks or domain. Most of them show some relevant contributions but they do not provide interoperability for all the possible types of platforms, systems and range of application.

The lack of a single standard of reference for interoperability of unmanned systems makes any choice difficult since it will have an impact one way or another on legacy platforms. However, some alternatives may fit better for a given set of requirements. Harmonizing the existing standards, by combining them into one, or by proposing a brand new standard, would obviously solve most of the problems, but it would have serious implications both in industry and other programs that have adopted them as their standard [13]. This is clearly beyond the possibilities of the ICARUS project on itself.

Along the studies, two candidates stood out from the rest, STANAG 4586 [14] and others related, and the Joint Architecture for Unmanned Systems—JAUS [16]. They are both stable, widely used and complete. STANAG pays a strong attention to the intelligence, surveillance and reconnaissance (ISR) data, while JAUS is instead more devoted to command and control interfaces of the platforms, robot navigation and perception.

In this context, both were created to address specific requirements in different domains. STANAG related standards are predominantly military and, even though they have been promoted for civil applications, their requirements are heavily demanding in terms of compliance. STANAG 4586 is mostly focused on UAVs, even though some other types of unmanned systems have been developed to meet this standard. It is perhaps very relevant for the interoperability of military assets across the different NATO members, but it is hard to be adopted by civil or research platforms without a strong investment. For instance, certifying a small multi-rotor UAV for the STANAG 7085 Interoperable Data Links for Imaging Systems is costly and probably a barrier for small platforms providers. Furthermore, the geographical constraints (NATO only), the focus on the bigger systems and the absence of open available implementation make this option less convenient. Likewise, JAUS was originally designed for UGVs. It is fair to say that JAUS has done great efforts to extend the coverage to any type of platform, and it currently considers any unmanned system as a generic asset in order to become truly multi-domain. Its root is also military, but it was soon transferred to the Society of Automotive Engineers (SAE International) where it is currently hosted.

According to our analysis, JAUS is fairly aligned with the needs of small unmanned platforms in terms of the interoperability described in Section 2. Also, JAUS has been successfully demonstrated in recent years for collaborative UAV-USV cooperative missions [15]. A quite direct traceability between ICARUS needs and the JAUS service sets is easily

derived. It is already compatible with popular transport protocols (TCP, UDP, serial) independent of the communication link beneath it, which makes it more flexible. And it is already multi-environment (air, ground and maritime). There exist both commercial and open source implementations. Unfortunately, there is a fee to access the JAUS documentation which may prevent some providers from using it. Nevertheless, the cost is deemed reasonable.

There are many other initiatives with strong support in different communities. According to the principles and needs for standardization defined above, these are considered software frameworks and middleware rather than full standards. For instance, the robotic operating systems (ROS) is used nowadays in many multi-robot systems. However, open-source initiatives are open and flexible by definition which may not provide the expected reference specification for future developments. These initiatives definitely add a lot of value to the development of small-unmanned systems, but they do not formally satisfy the interoperability requirements like the standards mentioned previously. They should remain at the platform level and the platforms should comply with an external interoperability standard. It is the scope of the interoperability work to harmonize this heterogeneity into a single standardized protocol.

4. Interoperability standard

The ICARUS standard interface for interoperability of heterogeneous fleets is based on the Joint Architecture for Unmanned Systems (JAUS) [16]. JAUS is a service-oriented architecture (SOA) that specifies a list of services commonly found in robotics. ICARUS interface describes the subset of standard messages that will be used in the ICARUS scenario and specifies all the details required to comply with the ICARUS interface.

4.1. Service sets

The interoperability interface is a service-oriented architecture (SoA). The most common services for unmanned systems interoperability are already defined in JAUS as a set of advanced standards. They are grouped as 'Service Sets'. The following ones will be used in ICARUS:

- Core Service Set (SAE AS5710 [17]): essential services such as transport, events, discovery, etc.
- Mobility Service Set (SAE AS6009 [18]): mobile platforms services.
- Environment Sensing Service Set (SAE AS6060 [19]): platform-independent sensor capabilities.
- Manipulator Service Set (SAE AS6057 [20]): platform-independent capabilities common across all serial manipulator types.

The concepts defined in the ICARUS data model can be matched against specific services in this architecture. **Figure 1** shows the specific services used in a real ICARUS operation.

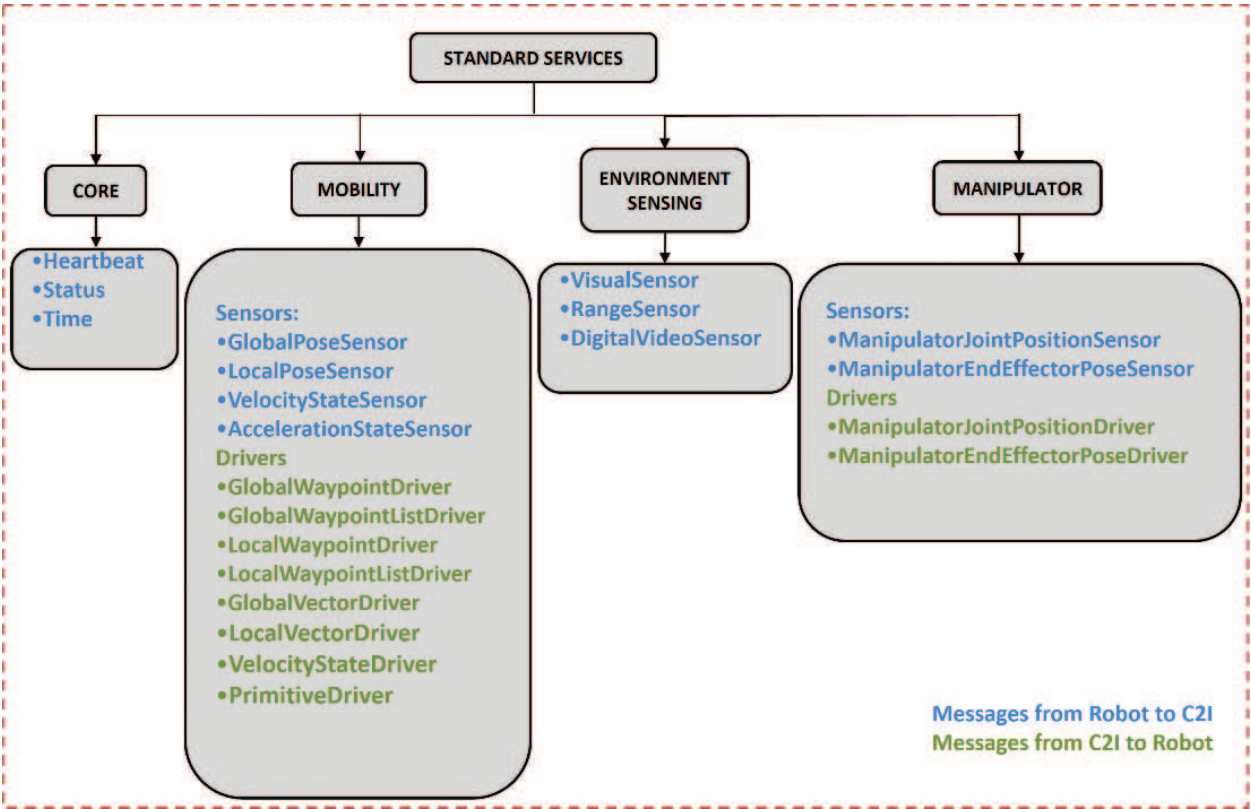


Figure 1. Relevant JAUS services (source: ICARUS).

However, as we progressed with all the integrations in the project, we discovered that some of the functionalities provided by some of the platforms were not supported by these standard services. We refer to this as the gap analysis. **Table 8** shows some of these gaps.

Given this, a new set of non-standard services has been defined to fill the gaps of the standard. This new non-standard service set is shown in the following **Figure 2**.

Gaps analysis				
	Outdoor quadrotors	Indoor quadrotors	Ground robots	Large sea vehicles
Survival kit deployment				
Rescue capsule deployment				
Platform-specific components enable/disable				
Platform extended status				
Manipulator tool selection				
Voice transmission				

Table 8. ICARUS interface gaps.

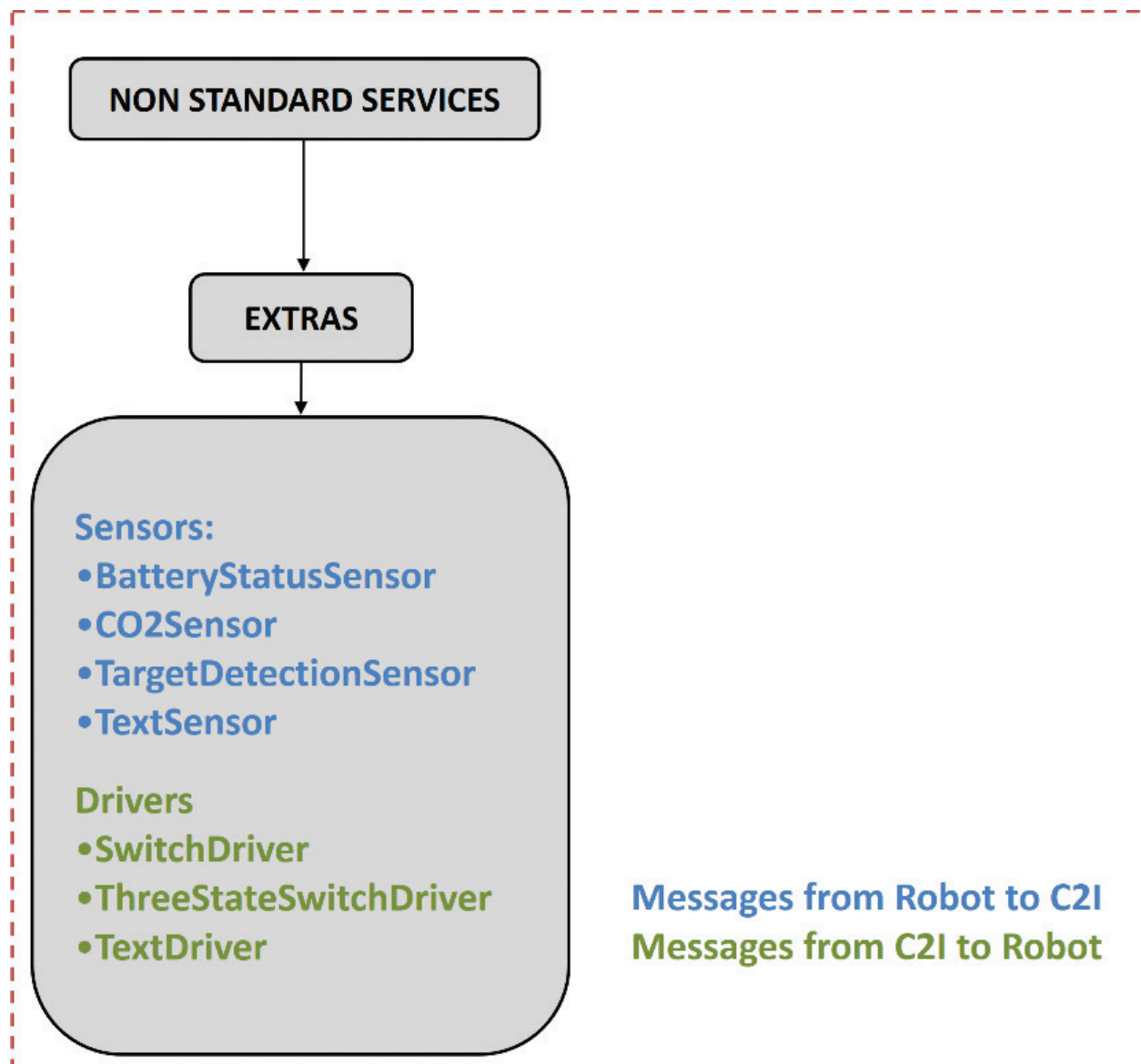


Figure 2. Additional non-standard JAUS services (source: ICARUS).

For each service, a strictly defined message-passing interface (vocabulary) and protocol (rules) for data exchange are available. There are generally three types of messages: query, report and command. Furthermore, the transport service (from the core service set) acts as an interface to the transport layer. Therefore, ICARUS interface is, in principle, independent from the physical transport layer. However, the current implementation for ICARUS is only available for the UDP protocol.

JAUS also defines a hierarchical and flexible topology built up of subsystems, nodes and components. For the implementation of JAUS within ICARUS, the following assumptions have been made:

- An ICARUS team is considered a system,

- Each platform is defined as a subsystem with a single node. Therefore, all components within the same platform will share the subsystem and node identifiers.
- As described later, a node may contain several components. But a component will implement only one service, plus the core services, which are always present. This restriction allows the C2I to dynamically discover each of the services available on each robot.

Therefore, an ICARUS system is depicted in **Figure 3**.

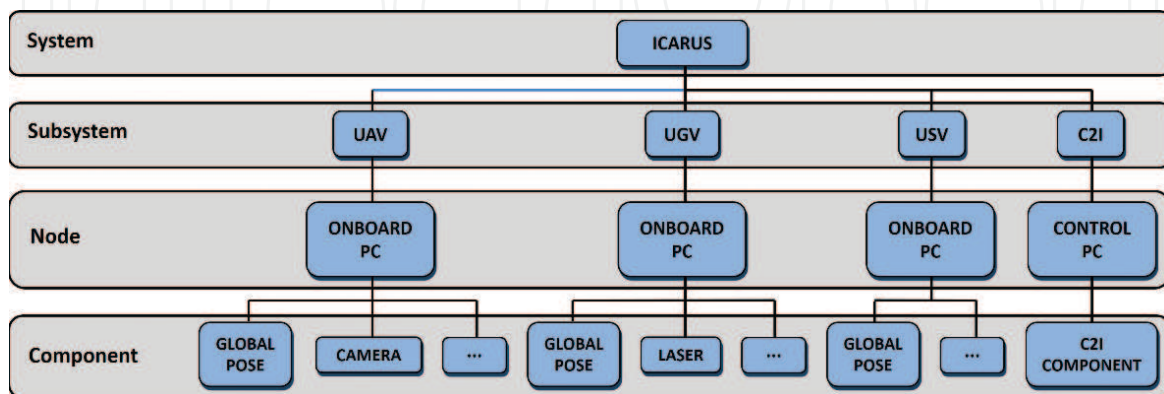


Figure 3. ICARUS JAUS topology (source: ICARUS).

5. An interoperable layer: the ICARUS library

All this functionality is provided to the ICARUS robotics partners as a software library referred to as the ICARUS interoperability layer. This module acts as a bridge between their internal and external development frameworks. This interoperability layer is also responsible for the integration of the ICARUS communication network and the command and control station on each individual platform.

A set of C++ classes has been designed to integrate the vehicles into the ICARUS network. The JAUS-specific functionality has been encapsulated within them. To comply with the ICARUS interface, a system may directly integrate this library (native integration). However, most robotics systems nowadays are based on either proprietary or open-source middleware (such as ROS). To accommodate these systems into an ICARUS compliant network, an alternative is to implement an adapter to the robot-specific middleware (translator). The diagram in **Figure 4** illustrates both cases, native integration (Robot C) and through an adapter (Robot A using ROS, and Robot B using MOOS).

The following sections describe the software classes encapsulated within the library and depicted in the previous diagram.

5.1. JAUS robot

JAUS robot encapsulates all the functionality required on-board the vehicle. It represents a subsystem in the JAUS topology (see **Figure 3**). In the ICARUS JAUS interface, a subsystem will

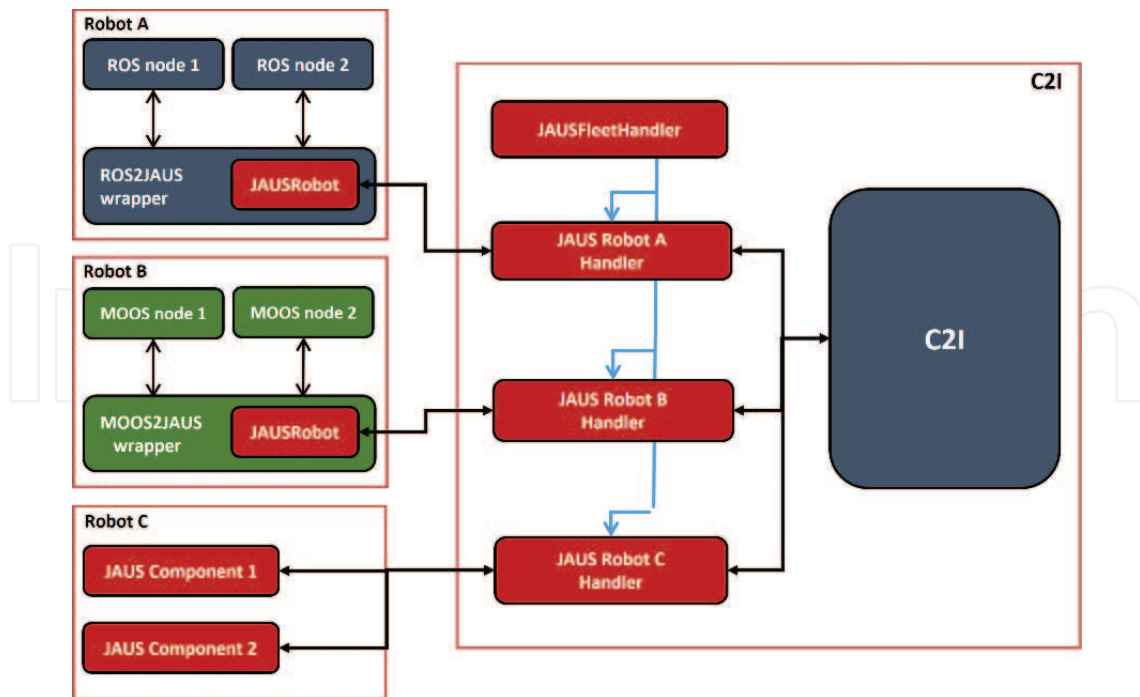


Figure 4. Robot adaptation strategy (source: ICARUS).

contain only one node. This approach will allow us to provide a component name to each service. Therefore, all components within the same robot share the subsystem and node identifiers. There are two types of services available on a JAUS robot in addition to the core services: sensors and drivers.

Sensors provide access to information generated on the robot (i.e. global pose, image, etc.). There are essentially two types of C++ functions required to integrate this functionality:

- Add sensor: a JAUS component is added to robot subsystem. For example, if the instance of JAUSRobot is myRobot, the following statement adds a GlobalPoseSensor service to our robot:

```
JAUSRobot myRobot;
myRobot.AddGlobalPoseSensor('OEMStar_GPS', AT_1_HZ);
```

- Set data: updates the data associated to a service. For the example above, the following lines will update the current GlobalPose of myRobot:

```
JAUS::GlobalPose globalPose;
myRobot.globalPoseSensor->SetGlobalPose(newGlobalPose);
```

Drivers, on the other hand, provide access to actuation capabilities provided by each robot (i.e. go to waypoint). There is an equivalent function required to integrate this functionality:

- Add driver: a JAUS component is added to the robot subsystem. For example, if the instance of JAUSRobot is myRobot the following line is adding a AddGlobalWaypointDriver service to receive waypoints request in global coordinate frame:


```
JAUSRobot myRobot;

myRobot.AddGlobalWaypointDriver('global_waypoints', authority_code);
```

The authority code parameter of these services is used for pre-emption and needs to be set lower to the one of the client accessing the driver. Otherwise, commands from the client are ignored.

Therefore, JAUSRobot creates a JAUS component for every new Sensor and Driver. This allows the JAUSFleetHandler class to discover and manage each of them independently.

The ICARUS JAUS interface is based on callbacks for message reception. One more function will register a local callback in order to receive any message coming from the JAUS network:

```
void localProcessMessage(const JAUS::Message* message){ }

myRobot.RegisterJAUSMessageCallback(localProcessMessage);
```

5.2. JAUS C2I

On the C2I side, two classes have been designed:

5.2.1. JAUS fleet handler

JAUS fleet handler encapsulates all the functionality related to the fleet management. It includes the functionality to discover subsystems and services on the JAUS network and retrieve their names and their current status. For example, if the instance of JAUSFleetHandler is myFleet, the following line allows discovering all subsystems on the JAUS network and retrieving their services names:

```
myFleet.DiscoverFleet();
```

On the other hand, the following line also allows checking for system updates:

```
myFleet.RefreshFleet();
```

In terms of JAUS, it represents a basic JAUS component implementing the discovery service to retrieve the subsystems available in the network and their services.

5.2.2. JAUS robot handler

JAUS robot handler is responsible for managing a single robot. After the discovery process, an instance of this class must be created and configured. This class will interface directly to the JAUS robot.

In terms of JAUS, it represents a basic JAUS component. For each sensor service available on the real robot, it creates an event of type every change. This is the JAUS mechanism to configure the sensor service on-board the robot to send data for every new set. Periodic events will also be available in the future.

On the other hand, for each driver service available on the real robot, it configures the access control service needed to send commands.

5.2.3. Management of the ICARUS communications

ICARUS tackles interoperability at different levels. This chapter focuses on the interoperability layer. This is a software-defined protocol (SDP) that can run over any compatible communication layer underneath. However, ICARUS also addresses the interoperability at the communications level, which is further detailed in Chapter 6 of this book.

A tight cooperation between the interoperability and communications layer allows for a smart management of the network. The ICARUS communications are a cognitive self-organizing multi-node network. This layer exposes an interface to configure the required data flows, its priorities and other details. Given the current set and number of robots, sensors and the priorities for the mission assigned, the communications layer can assure a quality of service for each data stream.

This is traditionally preconfigured manually for each mission. In ICARUS, a tight integration between the interoperability and communication layers has enabled the dynamic self-configuration of the team. A team management node runs within the C2I and exploits the discovery mechanism to retrieve all robots and their capabilities. This information is transferred to the communication layers that organized the data flows based on a-priori defined priorities. For instance, telemetry and telecontrol streams are given the highest priority since they are safety critical. For each robot, a camera is given the medium priority and all other sensors are lower priority. This a-priori priority allocation depends on the number of robots and their characteristics.

These configuration capabilities are also exposed to the C2I and therefore to the operator. The coordinator can at any time change the priority levels, enable new sensors, disable sensors that are not required, etc. The user is also informed with the current status of the network and he is therefore able to change the configuration if the network is overloaded.

Eight messages have been defined to exchange all the information needed between both interfaces (COMMS and JAUS):

- Register message.
- Activation of a stream.
- Deactivation of a stream.
- Deactivation warning.
- Network status notification.
- Robot's register notification. This is a notification sent from the COMMS interface to the JAUS interface. It is used to ask for the robot's register message presented before. It is needed when, for instance, the COMMS interface starts running later than the JAUS interface.

- Robot's unregister notification (COMMS to JAUS). It is sent when the COMMS interface loses a robot.
- Robot's unregister notification (JAUS to COMMS). It is sent when the JAUS interface loses a robot.

6. Individual platform adaptations

All ICARUS platforms have been adapted to the ICARUS interface. This automatically ensures the compatibility with the ICARUS C2I, enabling the multi-robot coordination and combination of data as later described in this chapter.

6.1. ROS to ICARUS robot node

All aerial platforms within the ICARUS project share a similar approach when it comes to software and hardware design. The hardware setup comprises a low-level board, responsible for the flight control of the vehicle (i.e. autopilot) and, optionally, a high-level board (i.e. on-board pc) responsible for the autonomous navigation and payload data. These autopilots communicate with the vehicle-specific ground station through the MAVLink protocol [21]. The on-board PCs, instead, run the robot operating system (ROS). A template has been developed to integrate ROS-based platforms. Therefore, all of them have been adapted using this template.

This template however should be configured to the specific characteristics of each platform, particularly in terms of sensors equipment. The proposed strategy is to implement a ROS-based wrapper to subscribe to ROS topics and interface the ICARUS protocol. This node is intended to run on-board the robot and provides a wrapper of ICARUS interface. The node is implemented within the robot.cpp file.

All the required services described in **Figure 1** are available for ROS-based systems through this template launch file. These services can be easily enabled by configuring a template ROS launch file. The XML excerpt below shows an example for the specific case of a global pose service:

```
<?xml version="1.0"?>

<launch>

  <node pkg="ros2jaus_node" type="robot" name="robot" output="screen">

    <!--ROBOT CONFIG -->

    <param name="subsystemName" type="string" value="ctae_robot"/>

    <param name="subsystemID" type="int" value="99"/>

  </node>

</launch>
```

```
<param name="nodeID" type="int" value="1"/>
<!-- GLOBAL POSE -->
<param name="globalPoseEnable" type="bool" value="true"/>
<param name="globalPoseSensorName" type="string" value="global_pose"/>
<param name="globalPoseUpdateRate" type="double" value="25"/>
<param name="globalPoseTopicName" value="/EURECAT_robot/global_pose"/>
</node>
</launch>
```

Therefore, a ROS-based robot just subscribes to a set of topics, with a predefined message type. An analysis of the existing ROS messages was performed to select the most appropriate interface definition. When an existing ROS message was deemed both valid and correct, this message was used. However, there were certain cases where the definition of the messages in ROS was either not existing, ambiguous or not valid. In these cases, a new message type was defined under the package `icarus_msgs`. The topic names and update rates can be configured from the ROS launch file:

- `/global_pose` (`icarus_msgs/GlobalPoseWithCovarianceStamped`)
- `/local_pose` (`geometry_msgs/PoseWithCovarianceStamped`)
- `/velocity_state` (`geometry_msgs/TwistWithCovarianceStamped`)

And publishes:

- `/local_waypoint` (`icarus_msgs/LocalWaypointStamped`)
- `/global_waypoint` (`icarus_msgs/GlobalWaypointStamped`)

6.2. Other robot adapters

The control systems of the ground platforms are implemented using FINROC, a middleware developed by the Robotics Research Lab at the University of Kaiserslautern. The rescue capsule runs OceanSys and exposes a data repository over which any software in the same network can subscribe and receive custom messages. ROAZ implements a similar system, but it is also ROS capable. The U-ranger on-board autonomy is developed in MOOS and implements a behavioural control strategy.

A template example was provided to each of the partners, together with the use case for ROS and some documentation. Each partner was able to quickly adapt its existing frameworks to the standard interoperable interface and become complying with all ICARUS team technology, such as the communication infrastructure and the C2I. **Tables 9 and 10** provide a summary of the ICARUS services provided by each system.

ICARUS services per robot								
SERVICE	AROT	ASOLAR	U-RANGER	ROAZII	UCAP	FIREFLY	SUGV	LUGV
Global pose sensor	Global pose (20 Hz)	Global pose (20 Hz)	Global pose (20 Hz)	Global pose (20 Hz)	Global pose (20 Hz)	Global pose (20 Hz)	Global pose (20 Hz)	Global pose (20 Hz)
Velocity state sensor		Velocity state (20 Hz)						
First camera	Left camera (20 Hz)	Visual camera (20 Hz)		Visual camera (20 Hz)	Visual camera (20 Hz)	Left camera (20 Hz)	Front camera (20 Hz)	Front camera (20 Hz)
Second camera	Right camera (20 Hz)	Thermal camera (20 Hz)		Thermal camera (20 Hz)		Right camera (20 Hz)	Manipul camera (20 Hz)	Manipul camera (20 Hz)
Third Camera	Thermal camera (20 Hz)					Thermal camera (20 Hz)	Rear camera (20 Hz)	Gripper camera (20 Hz)
Fourth camera	Visual camera (20 Hz)							
Range sensor (Point clouds)			Laser	Radar		Point cloud	Point cloud	Point cloud
Global waypoint list driver	Global waypoints	Global waypoints	Global waypoints	Global waypoints	Global waypoints		Global waypoints	Global waypoints
Primitive driver	Cmd velocity		Cmd velocity		Cmd velocity	Cmd velocity	Cmd velocity	Cmd velocity
Manipulator End effector pose sensor							End effector pose	End effector pose
Manipulator joint sensor							Joint position	Joint position
Manipulator end effector driver							End effector pose control	End effector pose control
Manipulator joint position driver							Joint position control	Joint position control

Table 9. ICARUS services provided by each vehicle.

ICARUS custom services per robot								
SERVICE	AROT	ASOLAR	U-RANGER	ROAZII	UCAP	FIREFLY	SUGV	LUGV
First switch driver	Delivery Kit				Deploy UCAP		Lights	Lights
Second switch driver					Inflate-raft		Manipulator	Manipulator
Third switch driver							Reset	Reset
Fourth switch driver							Audio	Engine
Fifth switch driver							Speech	Tool-Lock
First 3-state switch driver							Gripper control	Gripper control
Second 3-state switch driver								Tool selector
Text sensor							Text	
Text driver							Text Cmd	
CO2 sensor							CO2 sensor	
Robot extended status	Battery status	Battery status			Battery status		Battery status	Battery status
Video stream			Video stream					
Target detection	Victims	Victims		Victims			Victims	Victims

Table 10. ICARUS custom services provided by each vehicle.

7. Field validations: examples of multi-robot cooperation

The ICARUS approach to interoperability was initially verified with a set of in-lab integrations and simulated tests. Once the robotics platforms were finalized and ready for field tests, a series of field operations involving different combinations of pairs of air, ground and sea vehicles were organized during the integration trials carried out between July and September, 2014. The purpose was two-fold: (i) verification of the completeness, correctness and feasibility of the ICARUS interoperability interface; (ii) experimentation on the possibilities on multi-robot cooperative search and rescue missions. The results from one of these trials showed that the work on interoperability enabled large-scale cooperative mapping with multiple aerial and ground robots in urban search and rescue [22].

The final field validation was carried out together with the final integration and demonstration exercise of the ICARUS project described in Chapter 10. Three full-team validations were performed during the final project demonstrations:

- the maritime trials and demonstration in Alfeite, Lisbon (Portugal) in July 2015,
- the land trials and demonstration in Marche-en-Famenne (Belgium) in August 2015 and
- the participation in the euRathlon competition in September 2015 where the project received the Best Multi-Robot Coordination Award by the IEEE Robotics and Automation Society (RAS).

At this stage, all platforms had been integrated into the ICARUS system. After start-up, the current capabilities of the team could be dynamically discovered and the ICARUS C2I automatically configures itself, allowing an ICARUS team operator to plan a mission, assigning roles and tasks to each system. During the mission, all the information flows and the current network status are displayed. The operator can follow the progress of the mission, enable, disable or change the update rate of each of the ICARUS services. The operator can, at any time, request new missions, take manual control of the platforms that provide this service, resume previous missions, etc. All this functionality was exercised and demonstrated during the validations.

Together, these large-scale operational exercises completed the validation of the ICARUS interoperability standard interface. Therefore, ICARUS as a project has demonstrated multi-domain multi-robot heterogeneous interoperability in realistic search and rescue operations.

Some examples of the multi-robot collaboration experimented during ICARUS are described in the subsections below to illustrate the possibilities of multi-robot cooperation provided by an interoperable team.

7.1. Cooperative multi-stage aerial surveillance

In this multi-robot collaboration concept, the overall mission imposed by the human commanding officer is to provide a general assessment of a predefined sector. This is a typical scenario to be performed when relief agencies arrive on a crisis site. Assets to be deployed for this task are the fixed-wing UAV and the outdoor multi-rotor, both with clearly distinctive roles:

- The fixed wing aircraft acts as a surveyor system which covers the entire area quickly, flying at an altitude of around 100 m. Note that the altitude limitation is deliberative, as many countries impose a maximum flight altitude of 400 feet (133 m) for unmanned systems and it was our specific target to test the operational capabilities within realistic legislative bounds. Flying at this altitude, the aircraft quickly provides a general low-resolution assessment of the sector, such that areas of interest can be selected.
- The multi-rotor aircraft also acts as a surveyor system, but as it has a lower flight altitude of typically 40 m, it covers only a much smaller area. It is therefore used to provide a high-resolution assessment of an area of interest, as identified by the fixed-wing assessment.
- The multi-rotor aircraft also acts as observer system to provide high-resolution multi-view observation of points of interest (victims, buildings, etc.).

Operating multiple unmanned aerial systems in the same airspace is not easy from a safety perspective. In this case, vertical separation of the airspace was used for segregating the operations with the fixed wing and multi-rotor aircraft. The operators were also in constant contact with one another to synchronize the landing operations.

Figure 5 shows the different aerial systems in the air at the same time, whereas **Figure 6** shows the outcomes: the lower-resolution assessment by the fixed wing aircraft and the high-resolution assessment by the multi-rotor aircraft. As all data is geo-referenced, this information can be perfectly super-imposed on one another.



Figure 5. Multiple ICARUS aircraft in the air at the same time (source: ICARUS).

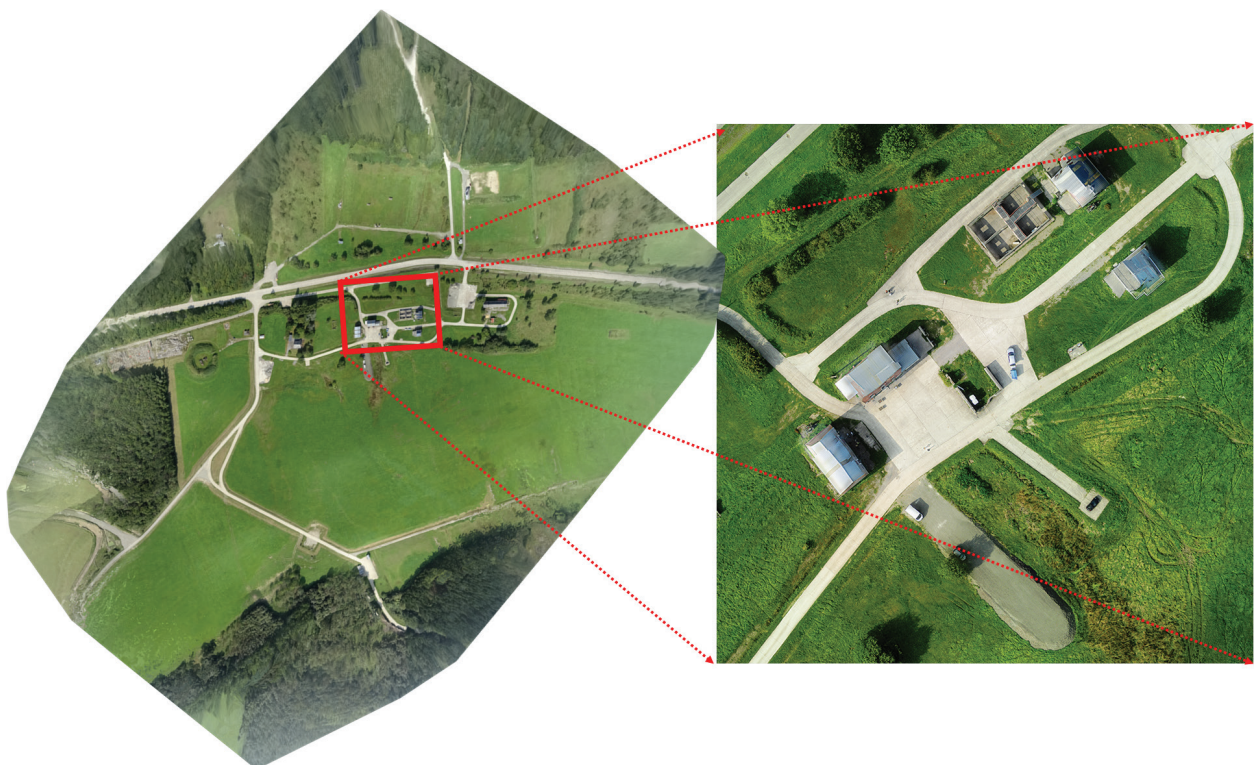


Figure 6. Low-resolution fixed wing assessment and high-resolution multi-rotor assessment (source: ICARUS).

7.2. Aerial scouting for traversability analysis

A common problem for relief teams is that the route they have to take due to their designated destination cannot be reached using the ‘normal’ routes, as roads are blocked by debris or by floods. The mission resulting from this use case is to use a multi-robot team to ensure the traversability of a route and provide an early identification of threats. The assets deployed for this mission are the fixed-wing and outdoor multi-rotor aircraft and the relief team on the ground, including ICARUS unmanned ground vehicles. The aircraft scans the area to detect blocked and cleared routes to the destination point and sends updated navigation information to the ground team, such that the ground team can travel to the destination as quickly as possible. **Figure 7** shows the ICARUS multi-rotor aircraft flying ahead of the ground team, searching for obstacles on the way to the destination.



Figure 7. ICARUS outdoor rotorcraft ensuring optimal routing of ground team (source: ICARUS).

7.3. Victim search

Victim search is a primary mission in any rescue operation. Following the typical mission profile, a search area is defined and a multi-robot collaborative search is ordered by the human commanding officer in this predefined area. Multiple collaboration modalities are possible, depending on the search and rescue context:

- In outdoor search and rescue scenarios, the fixed-wing and outdoor multi-rotor aircraft are deployed. The fixed wing aircraft can quickly provide a scan of a large area, either clearing the area or indicating preliminary detections, which then need to be confirmed by the outdoor multi-rotor aircraft. The latter then confirms the location and status of the detected

victims and can count the number of victims. This operation can take place in an urban search and rescue context, where victims are to be sought in rubble fields after an earthquake (as shown in **Figure 8**), or in a maritime search and rescue context, where victims have to be found in the water (as shown in **Figure 9**).

- In indoor urban search and rescue scenarios, the indoor multi-rotor aircraft and the small-unmanned ground vehicle are deployed to collaborative search for surviving victims inside semi-demolished buildings, as shown in **Figure 10**.
- In maritime search and rescue scenarios, the survival times in the water are often very short. Therefore, in an attempt to limit the time-delay between the search phase and the rescue phase in the relief operation, the fixed-wing and outdoor multi-rotor aircraft are deployed, together with the ICARUS unmanned surface vehicles and the unmanned capsules. This enables the unmanned aircraft to immediately steer the maritime vehicles towards the victims detected and localized by the aircraft. The unmanned surface vehicles also have sensors (infrared cameras) enabling victim detection on board, but as these are relatively small platforms, their field of view in rough sea conditions with high waves is limited. Collaborative victim search between aerial and marine platforms is therefore not impossible, but the greatest benefit of a mutual deployment lies in the combination of the search and the rescue aspect, as illustrated by **Figure 11**, which shows a victim in the water being tracked and localized by the outdoor multi-rotor aircraft, thereby guiding the unmanned surface vehicle to a position in the neighbourhood of the victim, such that an unmanned capsule can be deployed, which can inflate a life raft close to the victim in order to save the person.

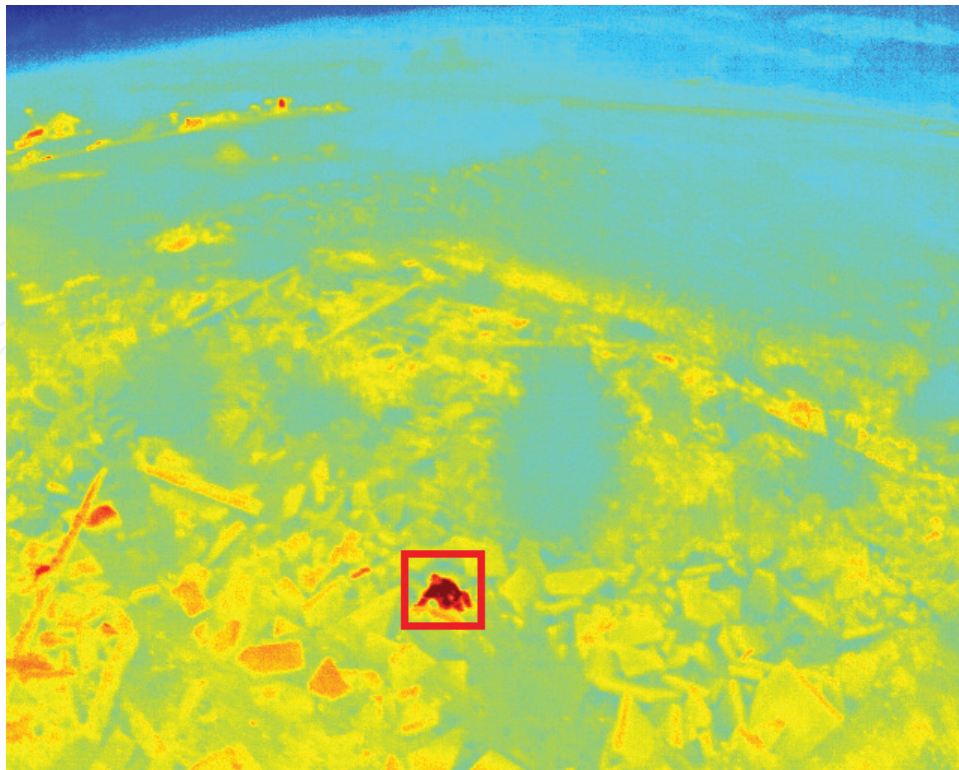


Figure 8. Automated victim detection on a rubble field in an urban search and rescue operation (source: ICARUS).



Figure 9. Victim in the water being localized by outdoor multi-rotor aircraft.



Figure 10. Collaborative indoor victim search (source: ICARUS).



Figure 11. Collaborative maritime victim search and rescue operation involving aerial and maritime platforms (source: ICARUS).

7.4. Carrier

During any search and rescue operation, many assets need to be deployed as quickly as possible. This mission profile shows how collaborative robotic agents can help by acting as carrier platforms for small assets and equipment and also for other unmanned systems. As an example, the large unmanned ground vehicle acts as a carrier platform for the small-unmanned ground vehicle and both the ROAZ II and the U-ranger act as a carrier for the rescue capsules. They not only enable the cargo to be transported to the destination, without any extra burden to the human relief workers, but also act as deployment systems for the smaller unmanned systems.

As an example, **Figure 12** shows how the large unmanned ground vehicle deploys the small-unmanned ground vehicle on the top of a building, whereas **Figure 13** shows how the rescue capsules are deployed from an ICARUS unmanned surface vehicle.



Figure 12. ICARUS large unmanned ground vehicle deploying the small unmanned ground vehicle on the top of a building (source: ICARUS).



Figure 13. ICARUS unmanned surface vehicle deploying the unmanned rescue capsule (source: ICARUS).

7.5. Communications relay

In the event of a large crisis, previously existing communication infrastructure is often broken or at least severely damaged. However, communication is crucial for having coordinated response operations. Collaborative unmanned systems can act as communication relay tools to extend the communication range over large distances. Of course, the assets which are most useful for this are the aerial tools, as they can provide line-of-sight communication relay over large distances. In the ICARUS project, an ad-hoc link-hopping network was developed, as detailed in Chapter 7 of this book, which allows to extend any communication link while the ICARUS aerial platforms are in the air. This allows the fixed wing aircraft and the outdoor multi-rotor aircraft to act as communication relays for the ground and marine rescue teams.

7.6. Air, sea, ground cooperation during euRathlon 2015

Only seldom, rescue operations have to be performed which span three domains (air, ground, marine). However, the Tohoku earthquake and subsequent Fukushima disaster showed that response protocols were ill prepared to approach such multi-domain crises. Therefore, the euRathlon challenge focussed specifically on this problem. In brief, the mission imposed by the euRathlon challenge consists of detection of, after a Fukushima-like incident, missing workers under water, outside on the ground and inside in a semi-demolished reactor building. Full information of the concept of operation is available online [23]. These search and rescue operations require the simultaneous and coordinated deployment of unmanned aerial, ground and underwater vehicles, gathering environmental data and performing real-time identification of critical hazards in a nuclear accident.

The ICARUS team deployed for this purpose five robotic systems:

- The outdoor multi-rotor was first deployed to search for the best route for the ground robots to reach the open entrance to the building. Then, it mapped the area in the RGB, gray and thermal spectrum. Finally, it performed real-time detection and localization of missing workers, leaks, etc.
- The Teodor UGV was used as carrier platform for the small UGV and for outdoor 3D mapping.
- The small UGV was used for indoor detection and localization of missing workers, leaks, etc.
- The ROAZ II vehicle was used as a carrier platform for the MARES unmanned underwater vehicle.
- The MARES unmanned underwater vehicle was used for underwater detection and localization of missing workers, leaks, etc.

Even though behaviours specific to euRathlon, such as opening valves were not originally considered in the ICARUS concept of operations, they were easily integrated as a proof of the flexibility of the followed approach towards interoperability. Thanks to the different levels of interoperability and automation, the specialized operator could take over at this point and tele-operate the system to finish the mission.

The following figures show the ICARUS team operating in the euRathlon scenario. **Figure 14** shows the ICARUS multi-rotor during his flight around the building, while **Figure 15** illustrates the Teodor UGV carrying the small UGV during the euRathlon challenge.



Figure 14. ICARUS rotorcraft during the euRathlon challenge (source: ICARUS).



Figure 15. Teodor and small UGV during the euRathlon challenge (source: ICARUS).

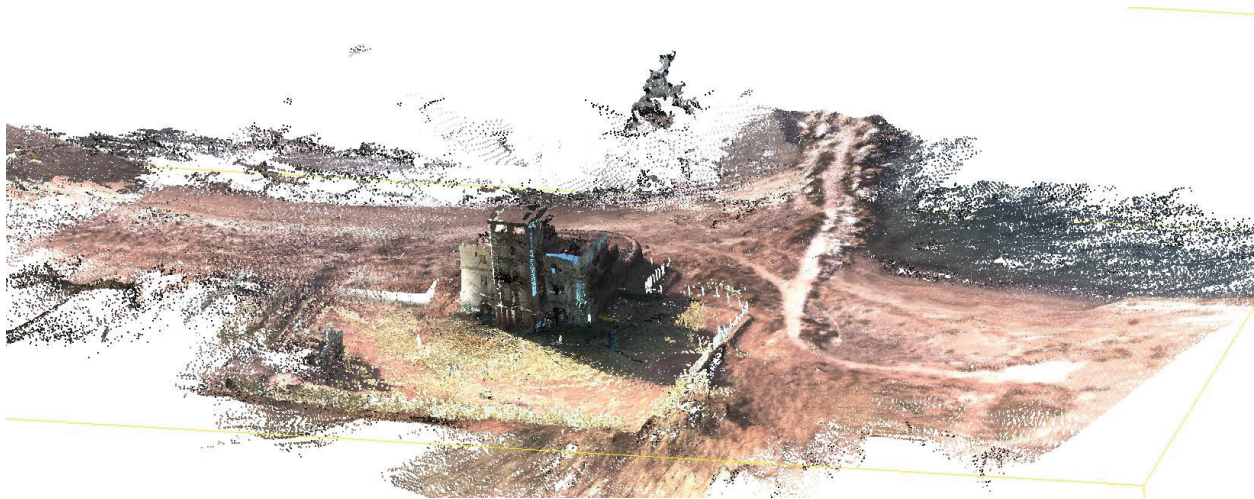


Figure 16. 3D map of the crisis area, obtained by combining 3D maps from the aerial and ground platforms (source: ICARUS).

Figure 16 shows the outcome of combining 3D maps obtained from the outdoor multi-rotor and ground platforms during the euRathlon challenge.

8. Conclusions

The work described in this chapter intended to integrate unmanned air, ground and sea vehicles developed by the different ICARUS partners into a heterogeneous fleet, collaborating as a coordinated, seamlessly-integrated team. A strong effort was devoted to appraise the existing body of work in standardization of multi-robot systems. Given the particular requirements of ICARUS, emphasis was placed on initiatives considering multiple domains (air, ground and sea). Likewise, given the platforms used in ICARUS, standards and methods applicable to smaller and lightweight platforms were prioritized. There have been several initiatives addressing both issues. However, harmonization of them is not yet a fact. There is still the need for a single multi-domain standard for interoperability, easily adaptable to both large and small systems. The contribution of the ICARUS project focused on the selection of the most appropriate existing initiative (JAUS), the evaluation of its application to multi-robot Search and Rescue missions, the elaboration of recommendations for improvements, the adaptation of all ICARUS robots and the demonstration of the ICARUS interoperable and heterogeneous team in three large-scale demonstration, exploring multi-robot cooperation and real-time centralized supervision and planning of an heterogeneous team.

Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement number 285417.

Author details

Daniel Serrano López^{1*}, German Moreno¹, Jose Cordero², Jose Sanchez², Shashank Govindaraj³, Mario Monteiro Marques⁴, Victor Lobo⁴, Stefano Fioravanti⁵, Alberto Grati⁵, Konrad Rudin⁶, Massimo Tosa⁷, Anibal Matos⁸, Andre Dias⁸, Alfredo Martins⁸, Janusz Bedkowski⁹, Haris Balta¹⁰ and Geert De Cubber¹⁰

*Address all correspondence to: daniel.serrano@eurecat.org

1 Eurecat Technology Center, Av. Universitat Autònoma, Cerdanyola del Vallès, Barcelona, Spain

2 Integrasys SA, Madrid, Spain

3 Space Applications Services NV, Zaventem, Belgium

4 Escola Naval, Rua Base Naval de Lisboa, Almada, Portugal

5 NATO STO Centre for Maritime Research and Experimentation, La Spezia, Italy

6 Eidgenoessische Technische Hochschule Zürich, Zürich, Switzerland

7 Technische Universität Kaiserslautern, Kaiserslautern, Germany

8 INESC TEC – Instituto de Engenharia de Sistemas e Computadores, Tecnologia e Ciência and Faculdade de Engenharia da Universidade do Porto, Campus da FEUP, Porto, Portugal

9 Instytut Maszyn Matematycznych, Warszawa, Poland

10 Department of Mechanical Engineering, Royal Military Academy of Belgium, Brussels, Belgium

References

- [1] Freeman, E., Robson, E., Sierra, K., & Bates, B. (2004). Head First design patterns. Sebastopol, CA: O'Reilly
- [2] Serrano D, Chrobocinski P, De Cubber G, Moore D, Leventakis G, Govindaraj S. ICARUS and DARIUS approaches towards interoperability. In: 8th IARP Workshop on Robotics for Risky Environment; Lisbon, Portugal; IARP January 2015, pp. 1-12
- [3] Serrano D. Key initiatives for interoperability through standardization. Applicability to small unmanned vehicles. In: The NATO STO Lecture Series SCI-271. 2015. NATO Science and Technical Organization – Educational Notes. DOI: 10.14339/STO-EN-SCI-271-01-pdf
- [4] Studer R, Benjamins VR, Fensel D. Knowledge engineering: Principles and methods. Data and Knowledge Engineering. 1998;25:161-197
- [5] Prestes E, Carbonera JL, Fiorini SR, Jorge VAM, Abel M, Madhavan R, Locoro A, Goncalves P, Barreto ME, Habib M, Chibani A, Gérard S, Amirat Y, Schlenoff C. Towards a core ontology for robotics and automation. Robotics and Autonomous Systems. 2013;61(11):1193-1204. DOI: <http://dx.doi.org/10.1016/j.robot.2013.04.005>

- [6] NATO. STANAG 4586, Edition No 2, Standard Interfaces of UAV Control System (UCS) for NATO UAV Interoperability – 2007
- [7] Andreas Birk and Max Pfingsthorn. 2006. A HMI supporting adjustable autonomy of rescue robots. In *RoboCup 2005*, Ansgar Bredenfeld, Adam Jacoff, Itsuki Noda, and Yasutake Takahashi (Eds.). Springer-Verlag, Berlin, Heidelberg 255-266
- [8] Goodrich M, Olsen D, Crandall JW, Palmer TJ. Experiments in adjustable autonomy. In: *IEEE Xplore Conference: Systems, Man, and Cybernetics, 2001 IEEE International Conference on*, Volume: 3
- [9] R. Parasuraman, T. B. Sheridan and C. D. Wickens, "A model for types and levels of human interaction with automation," in *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 30, no. 3, pp. 286-297, May 2000. DOI: 10.1109/3468.844354
- [10] Lacroix S, Alami R, Lemaire T, Hattenberger G, Gancet J. Decision making in multi-UAVs systems: Architecture and Ilgorithms. In: Ollero A, Maza I, editors. *Multiple Heterogenous Unmanned Aerial Vehicles*. Springer Tracts in Advanced Robotics ed. Berlin: Springer; 2007. pp. 15-48. DOI: 10.1007/978-3-540-73958-6_2
- [11] Farinelli A, Iocchi L, Nardi D. Multirobot systems: A classification focused on coordination. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*. 2004;**34**(5):2015-2028
- [12] ISO. ISO Standards Website – 2017 [Internet]. Available from: <http://www.iso.org/iso/home/standards.htm>
- [13] Pedersen J. Interoperability Standard Analysis (ISA). Standards Committee of the National Defense Industry Association (NDIA) Robotics Division; 2007
- [14] NATO. STANAG 4586, Edition No 3, Standard Interfaces of UAV Control System (UCS) for NATO UAV Interoperability – 2012
- [15] Incze ML, Sideleau SR, Gagner C, Pippin CA. Communication and collaboration of heterogeneous unmanned systems using the joint architecture for Unmanned Systems (JAUS) standards. In: *OCEANS 2015; 18-21 may 2015: Genova, Italy*. 2015. pp. 1-6
- [16] Rowe S, Wagner CR. An introduction to the joint architecture for unmanned systems. Technical Report, Ann Arbor, MI, USA. Cybernet Systems Cooperation
- [17] SAE International. JAUS Core Service Set [Internet]. 2008. Available from: <http://standards.sae.org/as5710/>
- [18] SAE International. JAUS Mobility Service Set [Internet]. 2009. Available from: <http://standards.sae.org/as6009/>
- [19] SAE International. JAUS Environment Sensing Service Set [Internet]. 2015. Available from: <http://standards.sae.org/as6060/>

- [20] SAE International. JAUS Manipulator Service Set [Internet]. 2011. Available from: <http://standards.sae.org/as6057/>
- [21] QGroundControl. MAVLink Micro Air Vehicle Communication Protocol [Internet]. 2016. Available from: <http://qgroundcontrol.org/mavlink/start>
- [22] Balta, H., Bedkowski, J., Govindaraj, S., Majek, K., Musialik, P., Serrano, D., Alexis, K., Siegwart, R. and De Cubber, G. (2017), Integrated Data Management for a Fleet of Search-and-rescue Robots. *J. Field Robotics*, **34**: 539-582. DOI:10.1002/rob.21651
- [23] Ferri G, Ferreira F, Djapic V, Petillot Y, Franco MP, Winfield A. The euRathlon 2015 grand challenge: The first outdoor multi-domain search and rescue robotics competition—A marine perspective. *Marine Technology Society Journal*. 2016;**50**(4):81-97

