

# Urdarbrunnen: Towards an AI-enabled mission system for Combat Search and Rescue operations

Ella Olsson<sup>1,\*</sup> Mikael Nilsson<sup>2,\*</sup> Kristoffer Bergman<sup>3,\*</sup> Daniel de Leng<sup>2,4,\*</sup>  
Stefan Carlén<sup>2</sup> Emil Karlsson<sup>2</sup> Bo Granbom<sup>2,†</sup>

**Abstract**—The Urdarbrunnen project is a Saab-led exploratory initiative that aims to develop an operator-assisted AI-enabled mission system for basic autonomous functions. In its first iteration, presented in this project paper, the system is designed to be capable of performing the search task of a combat search and rescue mission in a complex and dynamic environment, while providing basic human machine interaction support for remote operators. The system enables a team of agents to cooperatively plan and execute a search mission while also interfacing with the WARA-PS core system that allows human operators and other agents to monitor activities and interact with each other. The aim of the project is to develop the system iteratively, with each iteration incorporating feedback from simulations and real-world experiments. In future work, the capability of the system will be extended to incorporate additional tasks for other scenarios, making it a promising starting point for the integration of autonomous capabilities in a future air force.

## I. INTRODUCTION

Having rapidly progressed from expensive and custom-made hardware solutions to commercially-available off-the-shelf products, unmanned aerial vehicles (UAVs), colloquially referred to as ‘drones’, are becoming an increasingly common sight in today’s airspace. These vehicles are often part of unmanned aircraft systems (UAS) that can be used for a wide variety of tasks, both civil and military, ranging from camera shoots for movies and consumer product deliveries to improvised or specialized weapon platforms as observed in the Russian Federation’s ongoing invasion of Ukraine. The latter has shown that these type new technologies are crucial in today’s combat environment which motivates further long-term investments in innovative research [1].

The prevalence of UAVs in civilian applications is also resulting in the evolution of airspace management, with the European Union’s U-Space airspace initiative [2] expected to come into effect soon—thereby opening up novel opportunities for European business sectors—and autonomous airport solutions to accommodate them. On the other hand, the ease with which private actors can acquire and operate UAVs has resulted in new security challenges that also affect protected areas of national interest such as airports and power plants.

<sup>1</sup>Saab AB, Nettovägen 6, SE-175 41 Järfälla, Sweden.

<sup>2</sup>Saab AB, Bröderna Ugglas Gata, SE-581 88 Linköping, Sweden.

<sup>3</sup>RISE Research Institutes of Sweden AB, Fridtunagatan 41, SE-582 16 Linköping, Sweden.

<sup>4</sup>Department of Computer and Information Science, Linköping University, SE-581 83 Linköping, Sweden.

\*These authors contributed equally.

† Corresponding author: bo.granbom@saabgroup.com

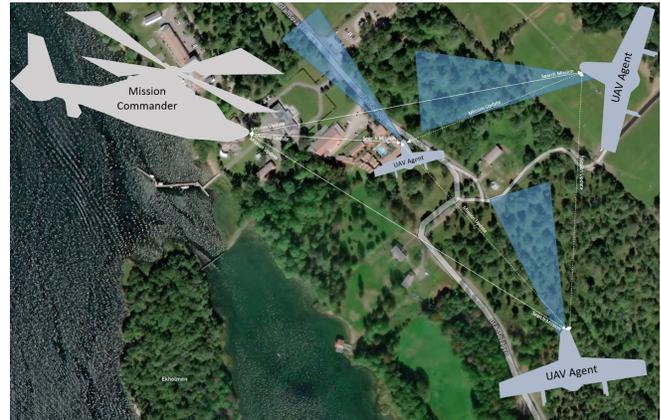


Fig. 1. An illustration of actors in a Combat Search and Rescue scenario, including a mission commander and a number of autonomous UAV agents performing a collaborative search. The background environment in the figure is courtesy of WARA-PS.

Saab is a Swedish security and defense company that strives to keep society and people safe [3]. In furtherance of this goal, Saab is the producer of defense and security products and services, including Saab’s Gripen fighter jet [4].

In this document we use the term Tactical Autonomy which we define as a technology that relate to functions that jointly and independently aim to fulfil a mission goal through the selection of different courses of action based on intrinsic knowledge and understanding of the situation and itself, as well as the predicted outcomes and associated constraints, such as risk acceptance, available resources, etc. Collaboration and teaming with human operators, as well as with other autonomous functions (multi-agent collaboration) is an essential part of the technology area.

There is an emergence of tactical autonomy solutions within the air domain, and in particular for UAS. It is envisioned that these solutions have the potential to drastically change the way security and defense are ensured. These systems have the potential of acting as a force multiplier in the areas of security and defense, especially in mixed teams consisting of autonomous agents and skilled humans. This is not a new idea either; UAS have already been demonstrated to be useful in civilian search and rescue (SAR) scenarios, with the *Hybrid Deliberative/Reactive* (HDRC3) framework [5] and related *WASP Research Arena for Public Safety* (WARA-PS) Core System [6] being among the pioneering research in that area.

This paper presents *Urdarbrunnen*<sup>5</sup>, which is an exploratory effort led by Saab towards developing a framework that can coordinate missions, i.e. a mission system, involving manned and unmanned systems operating in the air domain, with the goal of better understanding and learning about what kind of frameworks may be needed in the future. Concretely, this paper focuses on the first iteration of a planning, coordination and execution framework for mixed manned/unmanned missions. The overarching goal of the Urdarbrunnen project is to develop an initial architectural design that can be implemented and integrated in commercial off-the-shelf remotely piloted aircraft systems to provide full autonomy comparable to HDRC3, but in an operational domain in which it is able to complement and augment the capabilities of a modern air force. Following the example of SAR missions in the WASP Research Arena for Public Safety [6], we will focus on combat search and rescue (CSAR) missions in which a mission commander and autonomous UAV agents take on a (non-combat) supporting role; see Figure 1. CSAR missions differentiate themselves from SAR missions in a number of ways, including in the potential presence of an adversarial and disruptive force element, which make them a better fit for Urdarbrunnen.

The remainder of this paper is organized as follows. In Section II, we consider the place of autonomous UAS solutions in the military domain. Section III then presents the CSAR scenario which forms the basis of the Urdarbrunnen architecture design. The mission planning system underlying Urdarbrunnen is discussed in more detail in Section IV. An instantiation of the Urdarbrunnen architecture on a UAV Agent is presented in Section V, which is followed by a CSAR mission example in Section VI. We conclude the paper and look towards future work in Section VII.

## II. BACKGROUND

Contemporary autonomous systems depend on cognitive capabilities to monitor their surroundings, estimate the current state of the world, and predict what might happen next. They can make use of artificial intelligence (AI) algorithms and models in order to reason about, learn from, and interact with the world. Autonomous systems have been the subject of research and development activities globally for decades, at varying levels of complexity, and are consistently mentioned in strategy documents. For example, in 2005 the United States Department of Defense issued the “Unmanned Aircraft Systems Roadmap 2005–2030” [7], in which UAS are expected to possess various levels of autonomy. In addition, the Swedish commander-in-chief of the armed forces has expressed [8] an urgent need for e.g. autonomous capability development in the light of a coming NATO membership and current security policies.

Autonomous unmanned systems have a number of advantages over conventional manned systems or even remotely-piloted unmanned systems. They are 1) often smaller, allowing for operations in areas that are unsuitable for manned

platforms, 2) usually cheaper to develop, operate, and maintain, and 3) extending the operational domain to areas that are unreachable or unsuitable for manned platforms, hence can be used in situations that would otherwise be deemed too risky to a pilot or operator [9], [10]. Common tasks for these systems include supporting personnel on the ground or in the air through the delegation of high-level tasks, where AI methods are commonly used to break down and execute these tasks. Crucially, the collaboration between autonomous unmanned systems adds an additional layer of capabilities that utilize teams of systems of systems. This collaboration can be exemplified by multiple autonomous unmanned systems negotiating plans towards meeting an operator-set goal under a set of provided constraints. One example of a civilian system that is capable of collaborative planning is the HDRC3 system, which is used within both the WARA-PS research arena [6] and within the Autonomous Search System (AuSSys) [11] research project. Another example of a control architecture for controlling multiple UAVs for SAR in alpine scenarios is given in [12], where one human operator is able to coordinate the actions of multiple UAVs. A more detailed overview of recent work within AI-based mission planning for unmanned vehicles is found in the survey paper [13]. In the context of this paper, we instead focus on the military domain.

## III. SCENARIO

Many research results employ a Search and Rescue SAR style scenarios to demonstrate novel capabilities. In a similar vein, we adopt the CSAR [14] task and we let this task inspire our scenario of interest.

### A. Mission

As described in the United States’ CSAR Air Force Doctrine [14], a successful CSAR operation “enhances the Joint Force Commander’s (JFC) combat capability by returning personnel to areas under friendly control and denying adversaries the opportunity to exploit the intelligence and propaganda value of captured personnel.”. Whereas the primary objective of a CSAR mission is to recover isolated personnel such as downed aircrew, we will in the first phase of this project mainly focus on the initial phases where determining the location(s) of isolated personnel in an adversarial environment is of key importance. Therefore our main operational scenario focuses on the search part of the CSAR task, and we aim to employ search-capable fixed-wing UAVs for the search sub-task.

The scenario preparation consists of decomposing the CSAR task into a set of sub-tasks, namely a *Search*, a *Rescue* and a *Combat* sub-task—but as stated above, the focus in this development phase is on the Search sub-task. We situate the area of operations of this sub-task in the vicinity of Gränsö Castle, in Västervik motivated by the excellent research and demonstration opportunities available at this location, as well as the opportunity of being a part of the multi-domain community within the WARA-PS research program. WARA-PS [6] also offers the possibility to conduct mixed

<sup>5</sup>Based on Norse mythology: The Norns living near Urdarbrunnen were thought to determine a person’s fate.

academic/industrial research in multiple research areas, including but not limited to command and control of UAS in Search and Rescue missions, as well as offering excellent demonstration facilities in the Västervik area where research results can be presented and demonstrated. Therefore we align the search scenario—including its general environment where the scenario is situated in terms of i.e. topography, features, weather—with this location. This environment can be defined as a mix of open grassy terrain, leaf vegetation, shoreline and open water. We partly include open water for our task as the search operation may transverse from land to open water during the search. In the first phase of this project, we assume that the electromagnetic environment is non-obstructed, allowing us to exercise our communication links at full capacity. Furthermore, in this phase, we also assume that there are no hostile signal intelligence or communications intelligence present restricting the communication in relation of transmission and confidentiality aspects. Target intelligence includes one or more persons in distress, located anywhere in the vicinity of Gränsö castle, and we assume that this scenario instance (albeit unbeknownst to the agents) does not include any hostile agent threats, as the initial phase of this project mainly will focus on determining the isolated person(s) location. We aim to include threats in later phases of the project where we also will focus on the rescue and the combat support effort.

With regards to other intelligence objects we include our home base as a position for take-off and landing of our resources. In terms of cooperative forces, we include the ability to cooperate with external systems in the land, the maritime and the air domain. The purpose of this is to increase the operational effectiveness of our search operation. It might also exist neutral entities in the scenario that we must consider for safety reasons. Neutral entities may be civil boats or other vehicles, people, wild life etc. Our resources consists of a team of fixed-wing UAVs, each equipped with a pedestal mounted gimbal camera.

### B. Measures

We develop our version of the CSAR mission influenced by the CSAR task as defined in the United States Air Force Task List (AFTL) [15]. The task is listed under the capability PROVIDE PRECISION ENGAGEMENT within the framework for expressing the Air Force tasks, where PROVIDE PRECISION ENGAGEMENT is defined as “to command, control, and employ forces to cause discriminate strategic, operational or tactical effects.” [15, p. 87]. The CSAR task itself is described to includes capabilities “to organize, train, equip, provide, and plan for the conduct of prompt and sustained air operations to recover isolated personnel during wartime and contingency operations.” [15, p. 90]. Within the CSAR task we focus on the on two CSAR functions in particular:

- **AFT 2.3.1 PERFORM CSAR FUNCTIONS:** “To conduct operations to recover isolated personnel during wartime or contingency as necessary.” [15, p. 90].
- **AFT 2.3.4 PLAN CSAR FUNCTIONS:** “To consider all the particulars associated with the optimum utiliza-

TABLE I  
BREAKDOWN OF PERFORM CSAR FUNCTIONS.

<b>AFT 2.3.1 PERFORM CSAR FUNCTIONS</b>		
“To conduct operations to recover isolated personnel during wartime or contingency as necessary.”		
M1	Time	to recover distressed isolated personnel during wartime or contingency as necessary.
M2	Number	of personnel recovered during wartime or contingency operations.
M3	Percent	of successful CSAR operations.
M4	Cost	to perform CSAR functions.

TABLE II  
BREAKDOWN OF PLAN CSAR FUNCTIONS.

<b>AFT 2.3.4 PLAN CSAR FUNCTIONS</b>		
“To consider all the particulars associated with the optimum utilization of CSAR resources and to produce the necessary products to ensure effectiveness of CSAR functions is maximized.”		
M1	Percent	of resources used to conduct CSAR functions properly planned.
M2	Percent	of shortcomings in plans used to conduct CSAR functions.
M4	Time	to complete required planning to conduct CSAR functions.
M5	Cost	to plan CSAR functions.

tion of CSAR resources and to produce the necessary products to ensure effectiveness of CSAR functions is maximized.” [15, p. 91].

Based on these functions we develop the CSAR agent architecture as defined in and detailed in sections IV and V. We also adopt the corresponding measurements on a functional level as defined in the AFTL [15, p. 90-91] for these functions in order to perform an adequate evaluation of the mission. Inspired by the AFTL, we have selected the CSAR task as a *Mission Essential Task*. This has helped us to determine *what* to do, i.e. plan and execute the Search-part of a CSAR mission. We have also determined the conditions for this task by means of a scenario definition as detailed in Section III. The final step in developing our mission requirements involves selecting performance measures for the CSAR task as described in the AFTL [15, p. 64]. In this development phase, we omit the establishment of standards as also described in the AFTL [15, p. 64] as we at the moment of writing this paper, do not have the minimum acceptable proficiency required performance for the task at hand. The specific measures are selected from the AFTL [15, p. 90-91] and are detailed in Tables I and II.

## IV. URDARBRUNNEN PLANNING SYSTEM

In order for a system of autonomous agents to achieve complex goals, coordinated planning and execution of plans are essential capabilities. Autonomous agents must able to handle unexpected events during plan execution. Together these aspects require a tight coupling between planning and execution. It also requires any participating autonomous agent to be able to perform at least rudimentary local planning as far as it itself is concerned.

Planning in autonomous agents is facilitated by automated planning. Automated planning is a rich field within AI that

over the years has provided many different approaches to planning in many different domains. Research in automated planning has led to the development of a common planning language named *Planning Domain Definition Language* (PDDL) [16], [17], and many of the various planners available support planning in domains that make use of a subset of this language. Planners that can derive plans in any given domain are called *domain independent task planners*. They are often contrasted with *domain specific planners* that requires specific domain knowledge in order to plan in a given domain efficiently. The Urdarbrunnen planning system can leverage both types of planners depending on mission parameters.

Whenever agents interact it is important that their ontologies are aligned so that a concept like “flying” means the same to the agent performing the task and the agent planning it.

#### A. Abstraction level and planning approach

In order to provide a versatile architecture, capable of handling tasks of different complexity, the architecture should be capable of planning at different levels of abstractions. In a mission context, this naturally translates into being able to plan both centralized/globally and decentralized/locally. With centralized we mean that one planner derives the whole plan and with decentralized we mean that several planners provide smaller parts of a larger plan. Lower abstraction levels are suited for centralized planning and vice versa.

As an example of centralized low abstraction planning, a CSAR mission planner may plan almost every detail of each participant’s actions, e.g. detailed commands for TAKEOFF, FLY-TO and LOOK-AT actions. But the mission could also be planned in a decentralized way at a higher level of abstraction, letting the top-level planner stop at the level of SEARCH-AREA commands, leaving the agents to perform the decentralized planning of how to SEARCH-AREA by themselves.

The architecture allows for different levels of abstraction depending on mission requirements and command preferences.

#### B. Initiatives – top-down vs bottom-up approach

When using a lower level of abstraction, the centralized planner makes all important decisions. This is a purely top-down approach where agents are left with less room to take initiative and have less responsibility. In a bottom-up approach, in contrast, a planner may break down missions into tasks and further into sub-tasks that are published and distributed to participating agents. Agents can then by their own initiative reserve and perform tasks, being fully responsible for carrying them out. In the bottom-up approach, agents are also responsible for synchronizing tasks among themselves. In order to facilitate publication, distribution and synchronization among agents we add a *Blackboard* and a *Constraint Store* to the architecture.

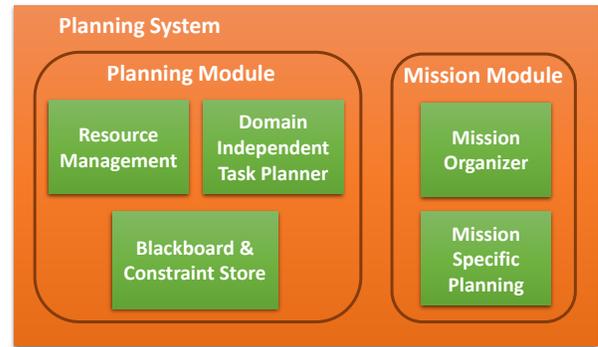


Fig. 2. Architecture of the Urdarbrunnen Planning System, consisting of a planning and a mission module.

#### C. Execution and re-planning

When a plan that solves the mission goal has been found, the next step is to execute it. In order to successfully execute a plan in the presence of unforeseen events, execution monitoring is needed. At the lowest level, an agent performing a task may need some kind of fallback, perhaps the execution follows a behavior tree model [18] that details what can go wrong and how to recover. If the agent cannot perform its task, plan execution enters a failure mode.

In a top-down architecture there is a global execution mechanism that when informed of the failure takes measures to repair the plan or come up with a new working plan.

A bottom-up architecture need to include a plan repair process that may involve first putting the failed task back on the blackboard and perhaps preventing the failing agent from reserving that task again after repeated failures.

#### D. The planning architecture

In order to meet the requirements discussed in previous sections, we define the planning system. An illustration of this system is found in Figure 2.

The planning system contains a mission independent *Planning Module* that contains the core planning capabilities that are needed to perform any mission. This is complemented by the mission-specific *Mission Module* that handles the mission-specific details for each type of mission that the system can perform.

The mission-independent planning modules are 1) *Resource Management*, keeping track of the systems resources and agent capabilities, 2) *Domain-Independent Task Planner*, required for planning and 3) *Blackboard & Constraint Store*, required to synchronize agents during missions.

The mission-dependent modules are the 1) *Mission Organizer*, responsible for putting together all aspects of the mission and executing it with the help of the 2) *Mission-Specific Planning*, containing all planning aspects that are outside of domain-independent task planning.

A concrete example of how to implement the planning system in a system capable of performing CSAR missions is given in the following sections.

## V. URDARBRUNNEN UAV AGENTS

This section presents the system architecture for the Urdarbrunnen UAV agents in terms of hardware, middleware and software. An overview of an agent is shown in Figure 3. Note that the figure shows all software modules. It is possible for agents to be part of the system without having all modules.

### A. Hardware

**Autopilot:** Pixhawk is an open-source hardware platform designed for the development of autonomous unmanned vehicles, such as drones, rovers, and other robotic platforms. It was first introduced in 2011 by the company 3D Robotics and has since become a popular choice among hobbyists, researchers, and commercial drone manufacturers. Pixhawk is compatible with various sensors, such as inertial measurement unit (IMU), Global Navigation Satellite Systems (GNSS), barometer and magnetometer, to provide a stable estimate of the physical state of the vehicle. The Pixhawk is also equipped with a micro controller that runs the firmware, which is responsible for controlling the motors, regulating the power supply, and communicating with other devices, such as a Ground Control Station (GCS).

**Companion Computer:** The UAV is also equipped with a companion computer. The companion computer is responsible for running the Robot Operating System 2 (ROS2) [19] software modules described in Section V-C, and is able to communicate with other UAV agents through the ROS2 network. The communication between the autopilot (Pixhawk) and the companion computer is done over Ethernet in order to minimize latency and maximize bandwidth.

**RC Transmitter:** The radio-control (RC) transmitter is used for remote control of the UAV by the safety pilot, whose main responsibility is to monitor flight and taking control of the vehicle if necessary to avoid any safety risks. Hence, the system must allow that a safety pilot is able to intervene.

### B. Middleware

The UAV agents use ROS2 as a middleware in order to communicate and share information. ROS2 is a distributed and modular software framework designed for building robotic systems. The new version is designed to address some of the limitations of the original ROS framework, including limitations related to scalability, real-time performance, and support for various hardware platforms. ROS2 also incorporates new features and improvements, including support for multiple operating systems and programming languages, a more modular architecture, and better support for real-time and safety-critical applications. One of the main components of ROS2 is the communication layer based on the Data Distribution Service (DDS) standard [20].

### C. Software

This section describes the Pixhawk autopilot software, as well as the different ROS2 modules running on the companion computer.

**PX4:** PX4 [21] is an open-source autopilot software developed specifically for the Pixhawk autopilots. It is a modular and highly configurable software stack that includes vehicle control, navigation, and mission planning functions. PX4 provides a flexible platform for the development and deployment of autonomous UAVs. It comes with support for various types of aircraft, including fixed-wing planes, multi-rotors, and Vertical TakeOff and Landing (VTOL) vehicles.

**UAV Module:** The UAV module is used to control the UAV and distribute information to other modules and agents. The bridging of messages between ROS2 and the PX4 software is done by connecting the microdds client in PX4 and a Micro XRCE-DDS Agent [22] in ROS2. The module contains the offboard flight controller, which bridges the flight control interface from the PX4 software into ROS2. The flight control component is thus responsible for exposing UAV-related ROS2 base actions such as TAKEOFF, LAND and FLY-TO. In swarm applications where tight coordinated control is required, such as formation flight, one could implement a flight control component in the UAV module that connects to and controls several UAVs simultaneously. The UAV module also contain components related to controlling the payload of the UAV, such as the camera. The camera control component exposes ROS2 APIs that can be used to perform actions such as TAKE-PHOTO, CONTROL-GIMBAL, RECORD-VIDEO and STREAM-VIDEO. Finally, the module contains a UAV information component, which main responsibility is to continuously distribute information about the state of the UAV (such as physical state, battery level, and flight-related information). The component is also aware of the different capabilities and attributes that are related to the UAV.

**Information Module:** The information module is used to collect and distribute all information that is relevant for the UAV agent. It provides a list of all capabilities and attributes that is associated with the UAV agent.

**Planning and Execution Modules:** Currently, the Urdarbrunnen UAV Agent uses the ROS2 based planning system PlanSys2 [23] to perform domain independent task planning. PlanSys2 enables easy handling of PDDL domains and problems by for instance facilitating incremental updates that reflect changes in the world. PlanSys2 is also capable of executing and monitoring the execution of derived plans. It relies however on external automated planners to do the actual planning. PlanSys2 is tested with the external planners POPF [24] and TFD [25], but any PDDL planner with the matching output format can be used, assuming it has a ROS2 integration. In the UAV Agent, the PlanSys2 PDDL executor is located in the Execution Module since this module can be used stand alone without the rest of the PlanSys2 system in a minimalist agent that relies on external planning. The rest of the PlanSys2 system is located in the PDDL Planner in the Planning Module.

The Resource Management module contains all available resources in the form of agent IDs and for each agent it also contains a list of capabilities belonging to it.

The Blackboard & Constraint Store is needed mainly

# Urdarbrunnen UAV Agent

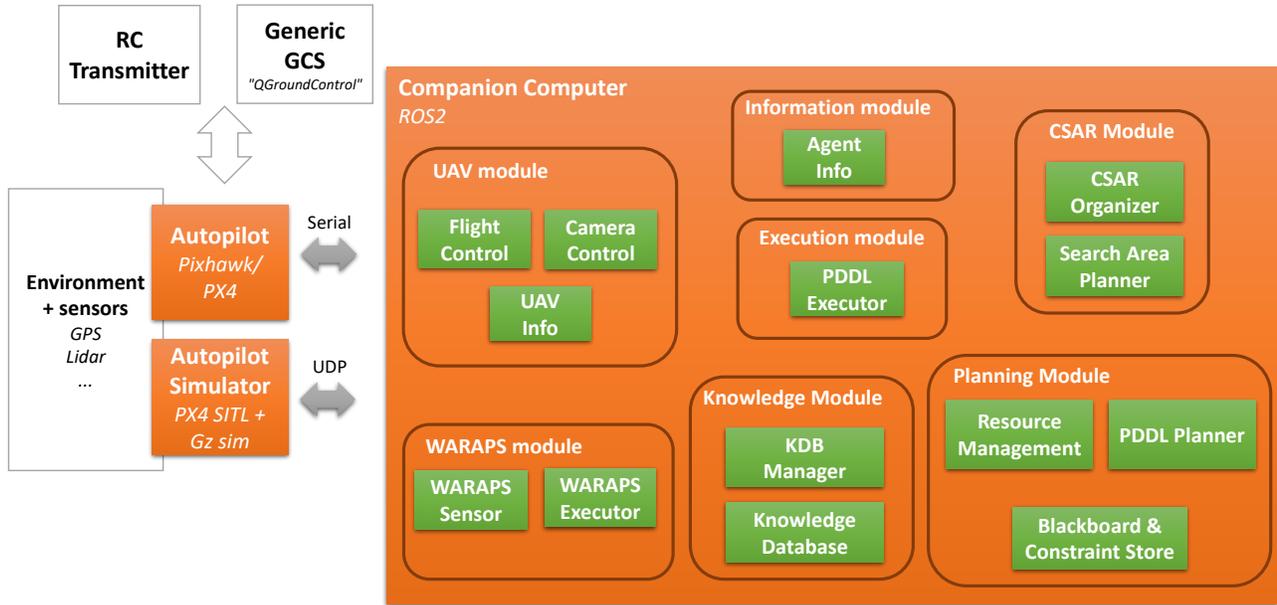


Fig. 3. Overarching view of the agent architecture, including hardware, (optional) software modules and interfaces.

when using a bottom-up initiative approach. When using a top-down approach the top-level planner is responsible for coordinating constraints between agents. However when performing a CSAR mission, the Blackboard can be used to communicate findings between agents so that agents can abort the search and return home when the missing persons are found.

**CSAR Module:** This is a mission specific module. It contains the CSAR Organizer that is capable of putting together a plan for a CSAR mission and executing it. It knows what type of resources are needed and how to put everything together. It is also responsible for monitoring the execution and re-planning from the current state in case something goes wrong. The first iteration of this module use a centralized top-down approach where one agent is responsible for deriving a detailed mission plan. In future iterations, we will compare different levels of agent initiative and task abstractions when solving the same type of missions.

The CSAR Module also contains the Search Area Planner, which is a mission specific planner. This planner support the CSAR Organizer by dividing the search area into sub areas that can be searched efficiently by a single UAV.

**Knowledge Module:** The knowledge module deals with the synchronization and combination of information at different levels of abstraction. It is responsible for storing and making accessible all types of knowledge ranging from high level semantic data down to low-level sensor data, such as stored images, LiDAR measurements, or behavioral models informed by doctrine. This information can be shared between agents in order to build up situational awareness. One

framework for knowledge representation is the SymbiCloud framework described in [26].

**WARA-PS Module:** The WARA-PS module contains components that are used to make the UAV agents compatible with the JSON API-specification for WARA-PS [6]. This module makes it possible to provide sensor information (physical state and camera stream) to the WARA-PS core system (Sensor Agent in [6]), but also the possibility to execute tasks (Direct Execution Agent in [6]) or even plan and delegate missions.

## D. Simulation

This section describes the simulation environment, which is used to perform initial test and validation of the ROS2 software modules.

**PX4 SITL:** With PX4 software-in-the-loop (SITL), it is possible to simulate the entire PX4 system, including the flight controller, sensors, and actuators, without performing real flights. This enables the possibility to test different configurations and settings of the autopilot prior to actual flights. PX4 SITL works by running the PX4 firmware on a virtual machine, which in turn communicates with a simulated environment through a network interface. The simulated environment can be a 3D robotics simulator, such as Gazebo or jMAVSim, or a custom simulator created by the user.

**Gazebo:** Gazebo [27] is an open-source 3D robotics simulator that allows for simulation of unmanned vehicles in various environments and scenarios. Gazebo provides a realistic simulation environment that can simulate the physics

of a UAV in flight. To simulate a UAV in Gazebo, one must first create a model of the UAV using so called simulation description format (SDF) files. These files define the physical properties of the UAV, such as its mass, size, shape, and aerodynamic properties. It is also possible to add and define sensors and other components of the UAV, such as camera, GNSS receiver, IMU, LiDAR, and motors. Once the model of the vehicle is created, it can be imported into Gazebo and placed in a virtual environment.

## VI. CSAR MISSION EXAMPLE

In this section we explain the messages sent within the UAS when performing a CSAR mission. The message sequence is illustrated in Figure 4. The system requires an external Tasking Authority that could be a human using a mission system. The Tasking Authority is responsible for populating the Resource Management component with available UAV agents and tasking the CSAR Organizer with the mission. The CSAR Organizer starts with allocating resources by asking the Resource Management for all agents with flying and camera capabilities. The CSAR Organizer then asks each agent for information regarding its search capabilities. This information is then forwarded to the Search Area Planner that decomposes the search area into smaller strips, each of which can be searched efficiently by a single agent. Using this decomposition, the CSAR Organizer puts together a PDDL problem file and sends this together with the CSAR PDDL domain file to PlanSys2 and receives a plan. The CSAR Organizer releases all agents that it will not make use of to the Resource Management and provides timing information for when the allocated agents will be used. It then starts executing the mission. Finally the CSAR Organizer reports that the mission is complete back to the Tasking Authority.

## VII. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented ongoing work from the first phase of the Urdarbrunnen project which aim is to iteratively develop an operator-assisted AI-enabled mission system for basic autonomous functions. In this first iteration, we have focused on finding relevant open-source, readily available components to swiftly and effectively, obtain an initial partial CSAR mission capability. This capability will be iteratively developed and it is now designed to be capable of performing a multi-agent search operation, in a complex and dynamic environment, while providing human machine interaction support for remote operators. Future work will involve simulations and real-world experiments in order to gain experimental results and to demonstrate the efficacy of the proposed system. As the aim the project is to develop this system iteratively, the capability of the system will be extended to incorporate additional tasks for other scenarios in future work. These scenarios may involve the integration of additional sensors and technologies, such as thermal cameras and LiDAR, to enhance the system's ability to detect objects and provide more accurate information to human operators. The system could also be extended to include more

decentralized planning, with agents capable of planning a task of higher level of abstraction by themselves. Another extension is to include machine learning models that enable the UAV agents to learn from experience. Additionally, further research could focus on the integration and evaluation of communication protocols to enable seamless collaboration and coordination among the drones and human operators.

## REFERENCES

- [1] FOI, "Totalförsvarets utmaningar på sikt kräver omfattande satsning på forskning idag." <https://www.foi.se/nyheter-och-press/nyheter/2023-03-02-totalforsvarets-utmaningar-pa-sikt-kraver-omfattande-satsning-pa-forskning-idag.html>, 2023. Last accessed March 2023.
- [2] European Union Aviation Safety Agency, "Commission implementing regulation (EU) 2021/664 of 22 April 2021 on a regulatory framework for the U-space." <https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32021R0664&from=EN>, 2021. Last accessed March 2023.
- [3] Saab AB, "Saab Purpose & Values." <https://www.saab.com/about/company-in-brief/purpose-and-values>, 2023. Last accessed March 2023.
- [4] Saab, "Fighter systems." <https://www.saab.com/products/air/fighter-systems>. Last accessed March 2023.
- [5] P. Doherty, J. Kvarnström, M. Wzorek, P. Rudol, F. Heintz, and G. Conte, "HDRC3 - A distributed hybrid deliberative/reactive architecture for unmanned aircraft systems," *Handbook of Unmanned Aerial Vehicles*, pp. 849–952, 2014.
- [6] O. Andersson, P. Doherty, M. Lager, J.-O. Lindh, L. Persson, E. A. Topp, J. Tordenlid, and B. Wahlberg, "WARA-PS: a research arena for public safety demonstrations and autonomous collaborative rescue robotics experimentation," *Autonomous Intelligent Systems*, vol. 1, pp. 1–31, 2021.
- [7] United States Department of Defense, "Unmanned Aircraft Systems (UAS) Roadmap, 2005–2030," 2005.
- [8] Swedish Armed Forces, "Överbefälhavarens råd avseende för- mågautveckling (FM2022-19979:13)," 2022. In Swedish only.
- [9] J. Gertler, "U.S. unmanned aerial systems," United States Congressional Research Service, 2012.
- [10] A. C. Watts, V. G. Ambrosia, and E. A. Hinkley, "Unmanned aircraft systems in remote sensing and scientific research: Classification and considerations of use," *Remote sensing*, vol. 4, no. 6, pp. 1671–1692, 2012.
- [11] Vinnova, "Autonomous search system, AuSSys." <https://www.vinnova.se/p/autonomous-search-system-aussys/>, 2022. Last accessed March 2023.
- [12] J. Cacace, A. Finzi, V. Lippiello, M. Furci, N. Mimmo, and L. Marconi, "A control architecture for multiple drones operated via multi-modal interaction in search & rescue mission," in *Proceedings of the 2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 233–239, IEEE, 2016.
- [13] A. Atyabi, S. MahmoudZadeh, and S. Nefti-Meziani, "Current advancements on autonomous mission planning and management systems: An AUV and UAV perspective," *Annual Reviews in Control*, vol. 46, pp. 196–215, 2018.
- [14] United States Air Force, "Combat Search and Rescue: Air Force Doctrine Document (AFDD) 2-1.6," 1998.
- [15] United States Air Force, "Air Force Task List (AFTL) Air Force Doctrine Document (AFDD) 1-1," 1998.
- [16] M. Ghallab, A. Howe, C. Knoblock, D. McDermott, A. Ram, M. Veloso, D. Weld, and D. Wilkins, "PDDL – the planning domain definition language (CVC TR-98-003/DCS TR-1165)," tech. rep., Yale Center for Computational Vision and Control, 1998.
- [17] M. Fox and D. Long, "PDDL2.1: An extension to PDDL for expressing temporal planning domains," *Journal of Artificial Intelligence Research (JAIR)*, vol. 20, pp. 61–124, 2003.
- [18] M. Colledanchise and P. Ögren, *Behavior trees in robotics and AI: An introduction*. CRC Press, 2018.
- [19] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot Operating System 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, pp. 1–12, 2022.

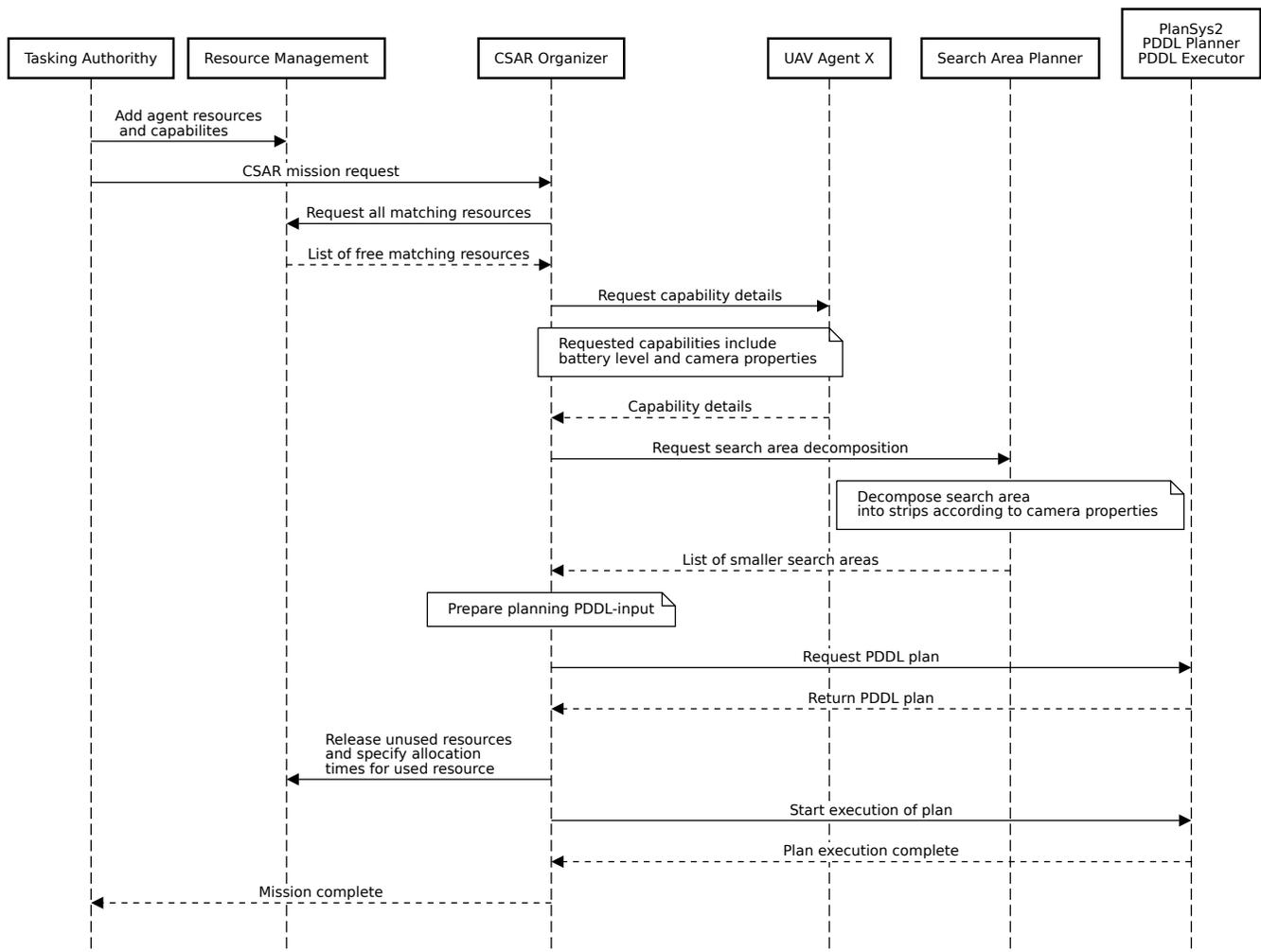


Fig. 4. Message sequence chart for a CSAR mission.

- [20] G. Pardo-Castellote, "OMG data-distribution service: Architectural overview," in *Proceedings of the 23rd IEEE International Conference on Distributed Computing Systems (ICDCS)*, pp. 200–206, IEEE, 2003.
- [21] L. Meier, D. Honegger, and M. Pollefeys, "PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms," in *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6235–6240, IEEE, 2015.
- [22] eProxima, "Micro XRCE-DDS agent." <https://github.com/eProxima/Micro-XRCE-DDS-Agent>, 2023. Last accessed March 2023.
- [23] F. Martín, J. G. Clavero, V. Matellán, and F. J. Rodríguez, "PlanSys2: A planning system framework for ROS2," in *Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9742–9749, IEEE, 2021.
- [24] A. Coles, A. Coles, M. Fox, and D. Long, "Forward-chaining partial-order planning," in *Proceedings of the Twentieth International Conference on Automated Planning and Scheduling (ICAPS)*, vol. 20, pp. 42–49, 2010.
- [25] P. Eyerich, R. Mattmüller, and G. Röger, "Using the context-enhanced additive heuristic for temporal and numeric planning," in *Towards Service Robots for Everyday Environments: Recent Advances in Designing Service Robots for Complex Tasks in Everyday Environments*, pp. 49–64, Springer, 2012.
- [26] P. Doherty, C. Berger, P. Rudol, and M. Wzorek, "Hastily formed knowledge networks and distributed situation awareness for collaborative robotics," *Autonomous Intelligent Systems*, vol. 1, pp. 1–29, 2021.
- [27] Open Robotics, "Gazebo." <https://gazebo.org/home>. Last accessed March 2023.