
SESAME
SECURE AND SAFE MULTI-ROBOT SYSTEMS

**Project Number 101017258**

# D2.1 Specification of MRS Capabilities

**Version 1.0**
**23 December 2021**
**Final**

**Public Distribution**

## University of Luxembourg

**Project Partners:** **Aero41, ATB, AVL, Bonn-Rhein-Sieg University, Cyprus Civil Defence, Domaine Kox, FORTH, Fraunhofer IESE, KIOS, KUKA Assembly & Test, Locomotec, Luxsense, The Open Group, Technology Transfer Systems, University of Hull, University of Luxembourg, University of York**

Every effort has been made to ensure that all statements and information contained herein are accurate, however the SESAME Project Partners accept no liability for any error or omission in the same.

© 2021 Copyright in this document remains vested in the SESAME Project Partners.

# Project Partner Contact Information


| | |
|---|---|
| **Aero41**<br>Frédéric Hemmeler<br>Chemin de Mornex 3<br>1003 Lausanne<br>Switzerland<br>E-mail: frederic.hemmeler@aero41.ch | **ATB**<br>Sebastian Scholze<br>Wiener Strasse 1<br>28359 Bremen<br>Germany<br>E-mail: scholze@atb-bremen.de |
| **AVL**<br>Martin Weinzerl<br>Hans-List-Platz 1<br>8020 Graz<br>Austria<br>E-mail: martin.weinzerl@avl.com | **Bonn-Rhein-Sieg University**<br>Nico Hochgeschwender<br>Grantham-Allee 20<br>53757 Sankt Augustin<br>Germany<br>E-mail: nico.hochgeschwender@h-brs.de |
| **Cyprus Civil Defence**<br>Eftychia Stokkou<br>Cyprus Ministry of Interior<br>1453 Lefkosia<br>Cyprus<br>E-mail: estokkou@cd.moi.gov.cy | **Domaine Kox**<br>Corinne Kox<br>6 Rue des Prés<br>5561 Remich<br>Luxembourg<br>E-mail: corinne@domainekox.lu |
| **FORTH**<br>Sotiris Ioannidis<br>N Plastira Str 100<br>70013 Heraklion<br>Greece<br>E-mail: sotiris@ics.forth.gr | **Fraunhofer IESE**<br>Daniel Schneider<br>Fraunhofer-Platz 1<br>67663 Kaiserslautern<br>Germany<br>E-mail: daniel.schneider@iese.fraunhofer.de |
| **KIOS**<br>Maria Michael<br>1 Panepistimiou Avenue<br>2109 Aglatzia, Nicosia<br>Cyprus<br>E-mail: mmichael@ucy.ac.cy | **KUKA Assembly & Test**<br>Michael Laackmann<br>Uhthoffstrasse 1<br>28757 Bremen<br>Germany<br>E-mail: michael.laackmann@kuka.com |
| **Locomotec**<br>Sebastian Blumenthal<br>Bergiusstrasse 15<br>86199 Augsburg<br>Germany<br>E-mail: blumenthal@locomotec.com | **Luxsense**<br>Gilles Rock<br>85-87 Parc d'Activités<br>8303 Luxembourg<br>Luxembourg<br>E-mail: gilles.rock@luxsense.lu |
| **The Open Group**<br>Scott Hansen<br>Rond Point Schuman 6, 5$^{th}$ Floor<br>1040 Brussels<br>Belgium<br>E-mail: s.hansen@opengroup.org | **Technology Transfer Systems**<br>Paolo Pedrazzoli<br>Via Francesco d'Ovidio, 3<br>20131 Milano<br>Italy<br>E-mail: pedrazzoli@ttsnetwork.com |
| **University of Hull**<br>Yiannis Papadopoulos<br>Cottingham Road<br>Hull HU6 7TQ<br>United Kingdom<br>E-mail: y.i.papadopoulos@hull.ac.uk | **University of Luxembourg**<br>Miguel Olivares Mendez<br>2 Avenue de l'Universite<br>4365 Esch-sur-Alzette<br>Luxembourg<br>E-mail: miguel.olivaresmendez@uni.lu |
| **University of York**<br>Simos Gerasimou & Nicholas Matragkas<br>Deramore Lane<br>York YO10 5GH<br>United Kingdom<br>E-mail: simos.gerasimou@york.ac.uk<br>nicholas.matragkas@york.ac.uk | |

# Document Control

| Version | Status | Date |
|---|---|---|
| 0.1 | Document outline | 1 October 2021 |
| 0.2 | Initial draft | 14 October 2021 |
| 0.3 | First draft | 1 November 2021 |
| 0.4 | Internal reviews | 18 November 2021 |
| 0.5 | Internal reviews updates | 2 December 2021 |
| 0.6 | Final version for partner reviews | 16 December 2021 |
| 0.7 | Final version partner reviews updates | 20 December 2021 |
| 0.8 | Further final version partner reviews updates | 21 December 2021 |
| 0.9 | Final version | 22 December 2021 |
| 1.0 | QA review | 23 December 2021 |

# Table of Contents

# List of Figures

# List of Tables

# Acronyms

**AGV** Automated Guided Vehicle

**APM** Autonomous Pest Management

**BIC** Battery Innovation Centre

**CTL** Computation Tree Logic

**EU** European Union

**ExSce** Executable Scenario

**GB** Gearbox

**LTL** Linear Temporal Logic

**MDL** Mission Description Language

**MRS** Multi-Robot Systems

**PNPs** Petri Net Plans

**RIT** Rotary Indexing Table

**RobotML** Robotic Modeling Language

**RTH** Return To Home

**SESAME** Secure and Safe Multi-Robot Systems

**TML** Task-based Mission specification Language

**TOG** The Open Group

**UAV** Unmanned Aerial Vehicle

**UL** University of Luxembourg

**UV-C** Ultraviolet-C

**WP** Work Package

# Executive Summary

This deliverable reports the specification of multi-robot systems (MRS) capabilities in Task 2.1 of the secure and safe multi-robot systems (SESAME) project.

This deliverable proposes a novel concept of stratagem, which encapsulates the capabilities of individual robots and robotic teams. It describes the capabilities of the multi-robot systems for the research and development activities that will be carried out within the SESAME project. In the first place, a stratagem enables engineers to specify and configure the hardware, software, and required capabilities. By using stratagems, engineers can describe high-level robotic capabilities, skills, roles, and behaviors based on the type of the robot and its onboard sensing devices. In addition to individual robot capabilities, stratagems facilitate the definition of capabilities at the team level and team coordination and collaboration, such as task decomposition and allocation. These novel and reusable capabilities reinforce the multi-robot system's robustness deployable across many different multi-robot systems use case missions such as:

- Use Case 1: Dependable Multi-Robot Systems in the Battery Innovation Centre.
- Use Case 2: Disinfecting Hospital Environments using Robotic Teams.
- Use Case 3: Power Station Inspection using Autonomous Multi-Robot Systems.
- Use Case 4: Autonomous Pest Management in Viticulture.
- Use Case 5: Security Management of Multi-Robot Systems-Based Assembly Lines.

These capabilities are incorporated in the SESAME knowledge base. Based on this knowledge base capabilities, then the WP3 can specify Executable Scenarios (ExSce) and transform them into executable models for simulation or the real multi-robot systems (see Figure 3 in the SESAME project description [1]). To this end, SESAME will make advances in three types of multi-robot systems capabilities, i.e., collaborative perception and sensor fusion, perception-aware trajectory planning, and collaborative intelligence.

It also describes the generic and application-specific architectural guidelines with a use case example of how to compose and configure capabilities of the multi-robot system into a more specific application that can instantiate for a diverse set of scenarios.

# 1 Introduction

European strategy and research roadmap documents emphasize the significant societal and economic benefits of robotic and autonomous systems. Multi-Robot Systems (MRS) comprise distributed and interconnected robotic teams that can carry out tasks beyond the competency of a single robot. Although, MRS offer improved scalability, performance, increased robustness, and mission enablement. The lack of a systematic engineering methodology covering the complete lifecycle of MRS results in solutions that fail because of fragile design and unrealistic assumptions. Secure and Safe Multi-Robot Systems (SESAME) project address these problems through an open, modular, model-based approach for the systematic engineering of dependable MRS. SESAME is underpinned by public meta-models, components, and configuration tools supporting the dependable MRS operation in uncertain settings characterized by emergent behaviors and possible cyber-attacks [1].

This document presents a stratagem to compose a MRS. SESAME will develop novel and reusable capabilities that reinforce MRS robustness deployable across many different MRS missions. Models of these capabilities are incorporated in the SESAME knowledge base. Thus, WP3 can be used to compose a MRS and an input to specify an Executable Scenario (ExSce) for simulation or the real MRS. To this end, SESAME will make advances in three types of capabilities, i.e., collaborative perception and sensor fusion, perception-aware trajectory planning, and collaborative intelligence.

## 1.1 Document Purpose

This document was prepared in the context of the European Union (EU) SESAME project. More precisely, it refers to Task 2.1: Specification of Multi-Robot Systems of Work Package (WP) 2: Sensor Fusion and Collaborative Intelligence. In particular, it is aimed to develop the novel concept of stratagem, by which to specify and configure the hardware, software, and the required capabilities. This contributes a set of generic and application-specific architectural guidelines to compose and configure the MRS capabilities into a larger application that can be instantiated for various use case scenarios.

## 1.2 Document Structure

This deliverable D2.1 document consists of the following sections:

- Section 2 discusses the related work in the specification and configuring of multi-robot systems capabilities in detail.
- Section 3 describes the specifications of multi-robot systems in terms of mission-level and robotic-level capabilities. Additionally, it explains the robot's skills, roles, and behaviors in detail.
- Section 4 describes the concept and methodology of stratagem with a sequence of steps on how a robotics engineer can select a robot according to the required capabilities, roles, and skills for a multi-robot systems mission-specific task.
- Section 5 describes the generic and application-specific architectural guidelines with a use case example of how to compose and configure capabilities of the multi-robot system into a more specific application that can instantiate for a diverse set of scenarios.
- Finally, Section 6 concludes this report.

## 1.3 Relationship to other Deliverables

This deliverable D2.1 provides the specification of multi-robot system capabilities based on the use case requirements in WP1, which have been specified in deliverable D1.1 – Project Requirements.

Based on the use case requirements, a generic stratagem methodology and guidelines are provided on composing and configuring the multi-robot system's capabilities into a more specific application. The dotted blue

rectangle in Figure 1 illustrates the focuses of the present deliverable D2.1. The system requirements, such as functional, non-functional, safety and, security requirements, are identified based on the capabilities. These capabilities and architectural guidelines thus set the knowledge base for executable scenarios to be specified in WP3. The deliverable D3.1 specification is represented by executable scenarios.

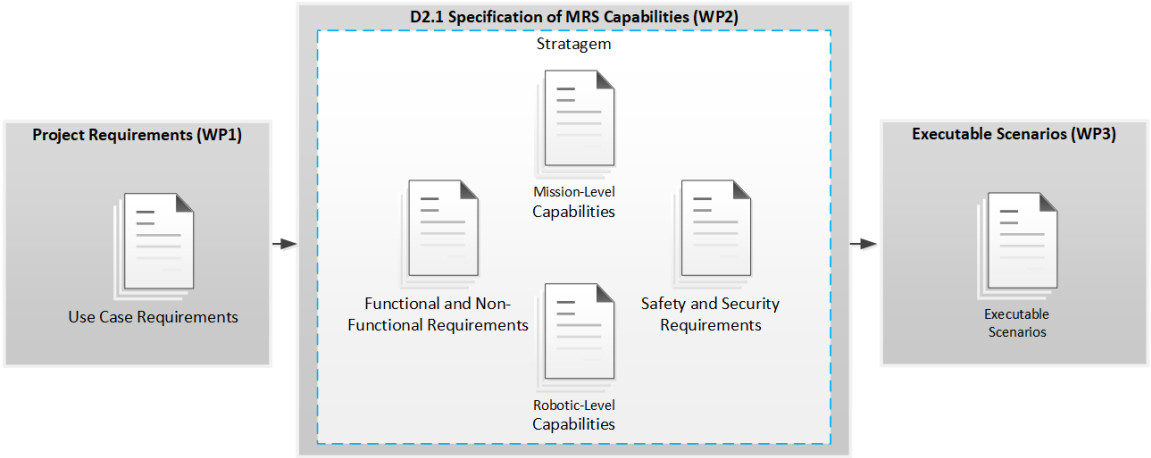


Figure 1: Workflow between the Project Requirements (WP1), Specification of MRS Capabilities (WP2), and Specification of Executable Scenarios (WP3). The dotted blue rectangle illustrates the focuses of this deliverable.

# 2 Related Work

This section discusses the related work in the specification and configuring of multi-robot systems capabilities in detail.

Several methods offer user-oriented ways of specifying and configuring multi-robot systems missions. These methods vary concerning the used language families and user interfaces as well as application domains, target platforms, and end-users. Prominent representatives for procedural multi-purpose programming languages are the swarm programming languages KARMA [2], VOLTRON [3], PaROS [4], and BUZZ [5]. In contrast, representatives for declarative markup languages include the Task-based Mission specification Language (TML) and the Mission Description Language (MDL) [6]. Programming-like languages provide a richer feature set and a higher degree of expressiveness than their markup-based counterparts. However, their handling also requires programming skills, making them less suitable for quick applications in dynamic ad-hoc scenarios. The end-user orientation of markup languages comes at the cost of reduced flexibility concerning the application domain and the used platforms. CommonLang and the PROMISE framework [7], which enables the specification of high-level goals rather than atomic mission steps, are further examples. Schwartz et al. [8] represent a mission in a flow-chart-like diagram that is easy to comprehend and simple to generate but lacks proper support for concurrency. Statecharts [9] have been successfully applied to describe robotic missions [10], [11], [12]. Statecharts are an extension of state machines and state diagrams, with a graphical syntax and formal semantics. This type of diagram requires a detailed definition of the steps a robot (or a team of robots) must perform in order to achieve a mission, which may become complicated in practical cases. Another formalism is Petri Nets [13], which have been successfully applied to develop robotic applications in the past [14]. In particular, Petri Net Plans (PNPs) [15] were developed to allow developers to describe plans for teams of robots. Despite their expressive power, PNPs require a precise definition of every action a robot is requested to perform. Furthermore, graphical user interfaces exist, typically easy to use but instead tightly bound to the application. There are several examples of such user interfaces for drones, including FLYAQ [16], the mission definition system [17], the FlightMaster software platform [18]. Robotic Modeling Language (RobotML) [19] enables the user to graphically model architecture, communications, behavior, and deployment of robotic systems as a further graphical approach.

Temporal logics, particularly Linear Temporal Logic (LTL) and Computation Tree Logic (CTL), are increasingly used by robotics experts for mission specification [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30]. LTL is also widely used due to the variety of existing planners which take LTL-based models as input. Furthermore, a temporal logic-based specification has other benefits as allowing users to analyze mission satisfaction through the use of model checkers. However, manually writing LTL specifications is complex and error-prone. To aid users in this process, mission specification patterns have been recently proposed to automatically generate mission specifications in temporal logics from recurrent mission specification problems [31]. Our proposed *stratagem* concept enables the creation of the *executable scenarios* through the definition of the MRS since they describe recurrent mission specifications that can be composed of complex missions. In order to compose such complex missions, we propose a set of *capabilities*. These *capabilities* allow composing independent robotic missions or collaborative missions. In our stratagem, we use domain expert's knowledge to rank the robotic-level capabilities and help in the decision making process to the robotic users when composing their MRS.

It is essential to understand the most relevant concepts to the following discussion of a domain analysis, namely *robot*, *component*, *action*, and *capability* [32]. A *robot* is defined as a physical agent that performs actions by manipulating the physical world and that has effectors and sensors. A *component* of a robot is defined as hardware, e.g., sensors and effectors, software control program or information object, e.g., maps and object models. An *action* is consciously, volitionally, and purposefully done by at least one actor. A *capability* is the ability to perform a specific action. A robot has a *capability* due to its components. It is difficult to compare a robot and an action directly because they have very different characteristics: a robot has physical, spatial characteristics whereas an action abstract, hierarchical ones. In order to be able to match between robots and actions the concept of "capabilities" is introduced. An action depends on a set of capabilities and a set of

components equips the robot with a certain capability. In order to exhibit a certain capability, a robot has to possess a set of components that cooperate and jointly enable the robot to exhibit the capability, is as shown in Figure 2 [32].

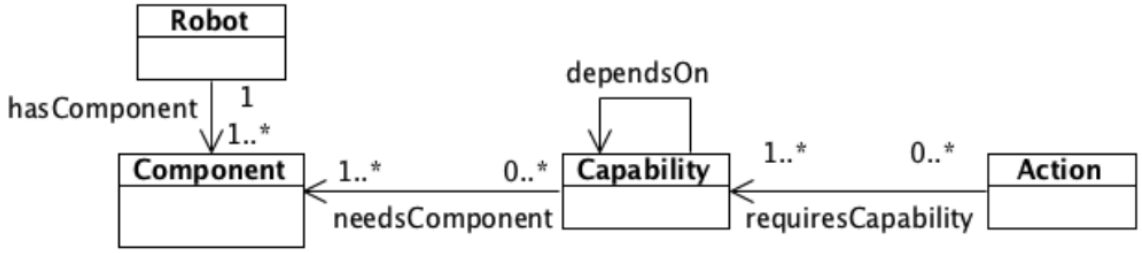


Figure 2: Relationship between the Concepts Robot, Component, Capability, and Action [32].

Parameterization is a component development technique in which components are adapted in many different ways by substituting the values of parameters which are embedded in the component. It allows codification of how adaptation is made within a component. The purpose of parameterization is to develop generic components that can be adapted in many ways by supplying values to parameters. This is not a new software engineering concept, and it has been implemented in many forms, such as subroutines, generics, macros, and preprocessors.

Identifying and understanding "factors" that result in different applications or different implementations of similar applications in a domain is the basis for parameterization. The factors considered in this method for parameterization can be largely classified into commonalities and differences of:

- The *capabilities* of applications in a domain from the end-user's perspective.
- The *operating environments* in which applications are used and operated.
- The application *domain technology* based on which requirements decisions are made.
- The *implementation techniques* (e.g., synchronization mechanisms used in the design).

Use case applications in a domain, although they provide a large set of common capabilities, provide different sets of capabilities, which make each application different from others. These capabilities from the perspective of end-users are modeled as features. A set of features characterizes the capability of the system. Therefore, features are used to parameterize domain products from the user's perspective.

In the development of a use case application, earlier decisions generally affect the range of decisions that can be made later, as shown in Figure 3 [33]. For instance, any prior decisions on an operating environment may affect the range of architecture models and implementation techniques that can be selected later. For example, in the viticulture domain, an Autonomous Pest Management (APM) system might be appropriate in a multi-tasking environment to identify the mission-level capabilities such as data collection, fungicide spraying, and monitoring the mission before development. The knowledge of inter-relationships between various development decisions is an important part of domain knowledge. Hence, we propose *capabilities-driven specification and configuration of multi-unmanned aerial vehicle missions for safe and secure multi-robot systems* based on the following capabilities:

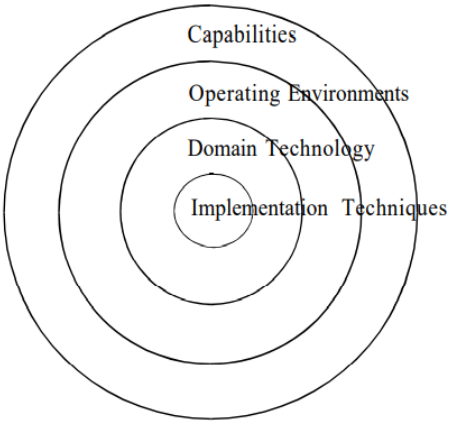


Figure 3: Types of Development Decisions [33].

# 3 Specification of Multi-Robot Systems

This section describes the specifications of MRS in terms of mission-level and robotic-level capabilities. Additionally, it explains the robot's skills, roles, and behaviors in detail.

Specification of MRS is at the core of the *stratagem*, which encapsulates the capabilities of the robotic team, individual robots, robotic roles, skills, and behaviours. SESAME develops novel and reusable capabilities that reinforce MRS robustness deployable across many different MRS missions. These capabilities are incorporated in the SESAME knowledge base. Figure 4 describes the high-level conceptual representation of the MRS specifications in terms of mission and robotic capabilities, skills, roles, and behaviors. The mission-level (see Section 3.1) and robotic-level capabilities (see Section 3.2) are represented in orange circles, skills (see Section 3.3) in grey circles, roles (see Section 3.4) in blue circles, behaviours (see Section 3.5) in brown circles, and the relationships are represented in directed arrows.

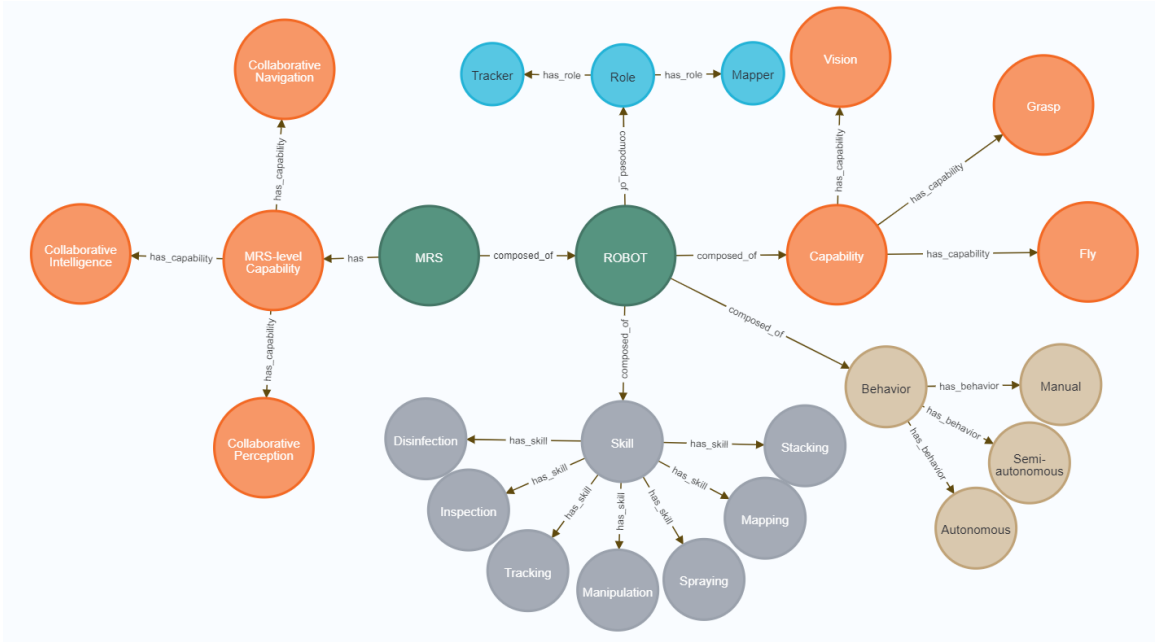


Figure 4: Specification of Multi-Robot Systems - Capabilities (Orange), Skills (Grey), Roles (Blue), Behaviours (Brown).

Table 1 represents the skill matrix, which defines the robot skills evaluation matrix based on key capabilities, associated roles, key components (sensors, software, hardware), and robotic behaviors for the various use cases. It is the key part of the stratagem. Scales ($5^*$: Excellent; $1^*$: Poor) in the matrix indicate the capability of the system to develop to execute a skill. The skill matrix can be relatively simple, depending on the customer requirements from the stratagem (see Section 4) methodology and the resulting system to develop. The advantage of having a skill matrix is the ability to implement the system's functionality in a step-wise manner. Doing so increases the timeliness and cost estimation reliability of the development process.

The following sources were used as inputs to the robot skills evaluation matrix based on key capabilities:

- **Domain experts**: People with extensive knowledge of robotic-level and mission-level complex systems and experienced users of these systems [34], [35], [36], [37], [38].
- **Domain literature**: Articles, papers, and evaluations of many different multi-robot systems [33], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48], [49], [50], [51], [52], [53].

The ranking should be consider as something dynamic. This means that the table should be updated as the TRL of the different technologies increases. As it was stated before, the values of the table will be populated by domain experts and robotic manufacturers according with the performance of the robotics executing the

different capabilities. In general, this skill aims to capture how good the robots are performing an specific operation.

Using the skill matrix (see Section 5), the engineer derives the system requirements (see Section 4) including functional and non-functional requirements, as well as functional safety and security requirements by meeting the customer's need. The system specification is represented by executable scenarios, which include the nominal behaviour of the use case, system boundaries, and logical scenarios (to be specified in WP3).

Table 1: Evaluation of Robot Skills based on Capabilities (5: Excellent, 1: Poor).

| **Skill-Based Capability Matrix** | | | | | | |
|---|---|---|---|---|---|---|
| **Robot Skills** | **Key Capabilities** | **Associated Role** | **Key Component** | **Robot Type** | | |
| | | | | **AGV** | **UAV** | **Robotic Arm** |
| Mapping | Mobility<br>Vision | Mapper | Perception<br>Mapping<br>Localization<br>Control | 3 | 5 | 1 |
| Detection | Vision | Detector | Perception | 4 | 2.5 | 5 |
| Tracking | Mobility<br>Vision | Tracker | Perception<br>Control | 3 | 4 | 2 |
| Manipulation | Mobility<br>Grasp<br>Vision | Manipulator | Perception<br>Localization<br>Manipulation<br>Control | 3 | 2 | 5 |
| Stacking | Mobility<br>Grasp<br>Vision | Stacker | Perception<br>Localization<br>Stacking<br>Control | 3 | 2 | 5 |
| Inspection | Mobility<br>Vision | Monitor | Perception<br>Mapping<br>Localization<br>Inspection<br>Control | 3.5 | 5 | 2 |
| Spraying | Mobility<br>Vision | Sprayer | Perception<br>Localization<br>Spraying<br>Control | 3.5 | 5 | 1 |
| Disinfection | Mobility<br>Vision | Monitor | Perception<br>Localization<br>Disinfection<br>Control | 5 | 2 | 1 |

## 3.1 Team-Level Capabilities

The SESAME project provides three types of capabilities at the robotic team (or MRS) level, i.e., collaborative perception and sensor fusion, perception-aware trajectory planning, and collaborative intelligence.

### 3.1.1 Collaborative Perception and Sensor Fusion

A collaborative perception for MRS uses multiple robots acting as viewpoints to improve and gather knowledge about the real environment. The knowledge could be in terms of obstacles, other robots, potential hazards or threats, and uncertainties exist in the environment. Collaborative MRS application scenarios require a rich, consistent and accurate understanding of the environment and robot state. While single robots frequently suffer from sensor limitations (e.g., range, occlusion), robotic teams can combine multiple observations and share their results. Therefore, the idea is to collect and fuse the sensor information from multiple robots using their individual perception capability, in order to improve the overall perception information of the environment. A single robot's situational awareness, as well as that of other robots, can be significantly improved with such accumulated perceptual skills. To exhibit the capability of perception, the robots need the on-board sensors such as internal sensors and external sensors, where the internal sensors include odometers, inertial navigation, and global positioning systems to determine the speed, position, and direction of the robot in the environment. On the other hand, the external sensors include infrared, laser, and vision, used to sense surrounding information, to extract the adequate feature information within the environment for processing and analysis, and finally to establish the environment model.

Perception is the primary capability enabling robots to reason and make decisions. Incomplete or unreliable information from perception components, e.g., due to failed sensors or limitations in their field of view, reduces significantly the ability of a robot to complete its mission successfully. MRS can mitigate this risk by fusing fragmented or partially correct information from robotic team members to collaboratively compose a more accurate representation of the environment, thus, supporting reliable reasoning and better decision-making. However, existing solutions for multi-source perception based on conventional computer vision techniques are overly resource-intensive, making them unfit for the processing capabilities of robots. Also, available perception solutions in robots lack precision and robustness due to inexact approximations and unrealistic simplifications. SESAME project aims to develop novel collaborative perception capabilities underpinned by robust object detection and pose estimation methods for localization and task execution, respectively, coupled with lightweight sensor fusion methods to accommodate the high uncertainty of sensor components. These capabilities will be variability-aware and fuse information from on-board sensors or peers based on runtime conditions, e.g., using RGB-D based cameras for moving objects to get detection information faster and with higher accuracy. Building on recent advances in deep learning and statistical machine learning, the perception capabilities will enable reasoning about the uncertainty of provided information, including real-time assessment of collected data, contributing to improved scene understanding.

### 3.1.2 Perception-Aware Trajectory Planning and Tracking

In this section, trajectory planner and tracker capabilities are introduced. Accordingly, the use cases should opt and specify components for the respective practice. Therefore, we tackle this as generic as possible.

Trajectory planning is one of the most important capabilities of SESAME to be addressed, i.e., to find an optimal path to the destination for high mobility of MRS [54]. Consequently, optimal decisions need to be taken for various mission-critical operations [55]. Accordingly, the trajectory can be defined as "*a time parameterized motion reference, i.e., geometric values of position, heading, derivatives associated with time law, passing through the waypoints*". One of the main objectives is to integrate perception-awareness into MRS trajectory planning, i.e., the construction of the trajectory on which the perception metric is incorporated [44], [56]. On the other hand, one of the main features of the trajectory planner is collision avoidance with other obstacles, humans, and other robots. These, generally, can be titled as static and dynamic obstacles [56]. So, we implement the updated information of the obstacles into the proposed planner. Furthermore, perception-awareness and safety/security are to be considered in the planning. Our main task falls at the planning level that we are going to address the above-mentioned aspects on versatile of MRS tasks. Another distinct feature to be implemented is the concept of situational awareness. This may include localization, geometric mapping, semantic mapping, obstacle detection and tracking, etc. So, we take into account the mathematical relation between

perception, sensing, and corresponding actions that MRS take. Indeed, it can be stated as collaborative and perception-aware trajectory planning, which minimizes environmental uncertainty by using data provided by other robots in a team combined with data collected by the perception components of the robot.

In addition to the trajectory planning, one significant capability is the trajectory tracking control, i.e., the design of the control commands to make the MRS be stable as well as to track the trajectory as close as possible [56]. The tracking task satisfies the desired position, velocity and orientation while fulfilling the given task. This can be defined as controlling the MRS to track the planned trajectory as close as possible. Also, the issue of stability is to be addressed. The latter is significant considering the model uncertainty. More importantly, the aspect we address is the structural design of planning and tracking parts, if we address and design them separately or simultaneously, considering the priority. This implies another preliminary step to be taken, as "*the general planning and tracking structural scheme, for the given MRS and use-cases, addressing the parts to be (de)centralized and on/offline*".

Finally, within the framework of autonomous robotic systems, the features of safety and security are vital. However, considering the available literature, the devised approaches are mainly stemmed from the corresponding definitions of safety and security, i.e., the definition implies the corresponding solution. The security can be the resilience of the solution to unauthorized access to communication channels. On the other hand, the safety can be interpreted as tolerance against a family of faults, avoidance from a given area, restriction of MRS states, and safely bound on the tracking drift. Cumulatively, we present the solution as reliable autonomous robotic systems with quality assurance, risk assessment and trust level [57], [58], [59], [60], [61], [62], [63].

### 3.1.3 Collaborative Intelligence

In addition to the above MRS capabilities, collaborative intelligence is another necessary mission-level robotic capability enabling the decomposition of complex tasks into smaller tasks manageable by individual robots, creating coalitions within an MRS, and task redistribution between team members. However, existing collaborative intelligence methods have limited applicability since they assume that the current knowledge of the MRS and the operating environment is sufficient to produce an optimal plan. SESAME aims at relaxing these assumptions using an experience-based planning method that extracts knowledge from previous similar plans by analyzing the divergence between the expected and actual plans and identifying potential sources for these deviations. Incorporating the extracted knowledge and the uncertainty experienced during plan executions into the planning process will enable the robotic team to jointly make better decisions and avoid highly sensitive plans that could put the mission at risk.

## 3.2 Robotic-Level Capabilities

The SESAME project provides three types of capabilities at the robotic level (or individual robots), i.e., mobility, grasp, and vision.

### 3.2.1 Mobility

Robots can have the mobility capability to move around in their environment and are not fixed to one physical location [64]. In the context of SESAME use cases, the following classes of mobility robots can be used:

- Fly (UAV)
- Ground Movement (AGV/Robotic Arm)

For example, the power station inspection use case has two operation modes, the normal operation mode, and the emergency operation mode, with different requirements. Normal operation mode is when no emergency event happens, but the power station needs to be inspected regularly for any anomaly detection. On the other

hand, during a fire event in Battery Innovation Center there is a requirement for a robot such as an Automated Guided Vehicle (AGV) with ground movement to transport the affected battery as soon as possible. An event such as an earthquake, tsunami, fire, or explosion occurs in the emergency operation mode. This operation has more strict requirements in the flying capability of the robot, such as an Unmanned Aerial Vehicle (UAV), and it is critical.

To exhibit the capability of flying or ground movement in the normal and emergency operation mode, the robot always depends on perception, mapping, localization, navigation, and control components.

### 3.2.2 Grasp

A robot's grasping capability is a key function to pick up and manipulate an object that helps people with complicated tasks. In order to provide daily support by using humanoid hands and arms, robots must have the ability to grasp a variety of unseen objects in human environments [65].

For example, within the Battery Innovation Centre (BIC) use case, robotic arms and AGVs capability during the production of high-voltage batteries (up to 800V). In particular, each robotic arm is mounted on a fixed station and takes part in the multi-step battery production process, including cell stacking, cell gluing, laser welding, and assembly. Although the use of the robotic team automates the battery production process, the production site for high voltage traction battery modules and packs is faced with the hazard of a thermal event of Li-Ion battery cells. Thermal camera sensors available within the BIC can detect the thermal event and trigger basic mitigation protocols under which the robotic arms and AGVs must change their working setting and collaborate to handle the burning Li-Ion cells. The aim is to get the burning hazardous material on the fastest way out of the production hall to the next safety zone to ensure the safety of workers and machines. Failing to dispose of the burning material safely or in time may have catastrophic consequences.

To exhibit the capability of grasping an object using a robotic arm, the robot always depends on perception, mapping, localization, navigation, and control components.

### 3.2.3 Vision

Vision refers to the capability of a robot to visually perceive the environment and use this information for the execution of various tasks [66]. Visual feedback enables robots to interact with the environment in various ways. In some cases, visual feedback is used for navigation and obstacle avoidance, while more complex examples include interaction with the user and manipulating objects. The simplest interaction between a robot and an object may be to, for example, push an object to retrieve information about the size or weight of the object. Here, simple visual cues providing approximate 3D position of the object may be sufficient. A more complex interaction may be to grasp the object to gain physical control over the object. Once the robot has the object in its hand, it can perform further actions, such as examining it from other views. Information obtained during interaction can update the robot's representations of objects and the world.

In the viticulture use case, the UAV must constantly monitor the environment and react autonomously when specific objects are detected in autonomous pest management.

To exhibit the capability of vision, the robot always depends on the camera sensor, object detection, and object tracking components.

## 3.3 Robot Skills

Skill defines how good a robot is while performing a specific role or capability. Skills enable tasks to become independent of the actual realization in components. A collection of skills is required for the robot to do a specific task. A component often implements a certain skill, but skills might also be realized by multiple

components. For example, a monitor robot in autonomous pest management requires perception, mapping, localization, navigation, inspection, and control components.

### 3.3.1 Mapping

A robotic mapping skill is to be able to construct a map and localize itself. Mapping in robotics is the procedure of creating a representation of the surrounding environment. To do that, a robot would use a variety of onboard sensors, like laser scanners, cameras, etc.

In the viticulture use case, a digital copy of the complete system of vineyards needs to be established and kept updated during the vegetative period of the year for autonomous mission planning. This digital vineyard includes datasets such as a high-resolution 3D model of the vineyard, including digital 3D vine models and plant physiology.

To exhibit the mapping skill, the UAV needs to carry visual and laser sensors to perceive the environment and collect the point clouds of the vineyard.

### 3.3.2 Detection

An object detection skill is identifying and locating all the obstacles in an environment.

In the viticulture use case, the UAV must constantly monitor the environment.

The UAV needs to carry visual sensors and object detection components to perceive the environment to exhibit object detection skills.

### 3.3.3 Tracking

An object tracking skill is looking at particular moving objects in real-time.

In the viticulture use case, the UAV must constantly monitor the environment and react autonomously when specific objects are detected.

The UAV needs to carry visual sensors and object tracking components to perceive the environment to exhibit object tracking skills.

### 3.3.4 Manipulation

Robotic manipulation skill refers to the ways robots interact with the objects around them: grasping an object and packing an order into a box.

For example, within the BIC use case, robotic arms and AGVs are used to get the burning hazardous material on the fastest way out of the production hall to the following safety zone to ensure the safety of workers and machines. Failing to dispose of the burning material safely or in time may have catastrophic consequences.

The robot always depends on perception, mapping, localization, navigation, manipulation, and control components to exhibit the manipulation skill.

### 3.3.5 Stacking

A robotic stacking skill is to pick up an object stacking on top of another object.

For example, within the Battery Innovation Centre, robotic arms and AGVs collaborate during the production of high-voltage batteries (up to 800V). In particular, each robotic arm is mounted on a fixed station and takes

part in the multi-step battery production process that includes cell stacking, cell gluing, laser welding, and assembly.

A robot requires a robotic arm and perception, mapping, localization, navigation, stacking, and control, including interacting with different kinds of objects, to exhibit stacking skills.

### 3.3.6 Inspection

A robotic inspection skill is to perform flaw or failure detection in a mission.

In the viticulture use case, the UAVs need to be deployed for real-time monitoring, surveillance, and coordination for data collection and spraying missions. These UAVs need to fly at least the same time that the spraying UAV to monitor the whole mission without being a bottleneck.

To exhibit these missions, the UAVs need to carry visual and laser sensors to perceive the environment, mapping, localization, navigation, inspection, and control components to evaluate the state of the mission.

### 3.3.7 Spraying

A robotic spraying skill is to navigate towards an infection hotspot and apply a predefined rate of fungicides.

In the viticulture use case, UAV needs to navigate towards an infection hotspot and apply a predefined variable rate of fungicides. This rate can vary depending on the grade of infection. Therefore, high importance is set on the adequate amount of fungicide applied to a specific hotspot.

The spraying UAV needs exact positioning accuracy to maintain a low but constant distance to the vines, mapping, localization, navigation, spraying, and control components to exhibit spraying skill. The efficiency and security of the fungicide spraying rely on high positioning accuracy, for example, due to the loss of RTK correction data.

### 3.3.8 Disinfection

Disinfection is an essential concern in-hospital care. In the disinfecting hospital environments using robotic teams use case, KELO ARODIS is a fully autonomous fleet of robots for disinfection of contact surfaces and aerosols. Ultraviolet-C (UV-C) is a known method to disinfect surfaces. Thus, the UV-C robot carries six 256nm UV-C lamps and autonomously moves to the surfaces to be disinfected. However, UV-C light imposes some risks for humans. In order to ensure that persons are not over-exposed with the maximum daily dose, a person detection system is installed. Once a person has detected the lamps turn off. Target customers are hospitals, public infrastructures like universities, trade fairs, and airports. They all have in common that they are either system-relevant public facilities or provide large buildings for people.

The disinfection robot depends on perception, mapping, localization, navigation, disinfection, and control components to exhibit the disinfection skill.

## 3.4 Robot Roles

A role is a specific task or activity associated with someone (individual or team). Someone that takes a particular role typically is an expert in a particular field (e.g., object recognition). It is associated with specific tasks, duties, rights, and permissions which do not overlap with other roles. A role has a specific view of the system at an adequate abstraction level using relevant elements only. A role is responsible for supplying a part of the system. “Role” in the sense of a participant of the system.

### 3.4.1 Detector

A robot's detector role is to detect and provide the classification of objects.

In the viticulture use case, the UAVs need to take flight for at least the same time as the data collection and spraying drone to monitor the whole mission without being a bottleneck. In the Battery Innovation Centre use case, the two robot arms work together to lift an object. To this end, the robots need the detect and classify the manipulated object.

The robot must carry visual sensors and object detection components to detect the other robots in the mission and exhibit the detector role.

### 3.4.2 Tracker

A robot's tracker role is to track the objects and provide the pose and velocity of objects.

In the viticulture use case, the UAVs need to take flight for at least the same time as the data collection and spraying drone to track the whole mission without being a bottleneck. In the Battery Innovation Centre use case, the two robot arms work together to lift an object. To this end, the robots need the 3D pose of the manipulated object.

The robot must carry visual sensors and object tracking components to track the other robots in the mission and exhibit the tracker role.

### 3.4.3 Mapper

A robot's mapper role is to construct a map and localize itself. The robotic mapping acquires a spatial model of a robot's environment. Maps are commonly used for robot navigation. To acquire a map, robots must possess sensors that enable it to perceive the outside world. Sensors commonly brought to bear for this task include cameras, range finders using sonar, laser, and GPS.

In the viticulture use case, a digital copy of the complete system of vineyards needs to be established and kept updated during the vegetative period of the year for autonomous mission planning. This digital vineyard includes data sets such as a high-resolution 3D model of the vineyard, including digital 3D vine models and plant physiology.

The robot needs to carry visual and laser sensors to perceive the environment and acquire a vineyard map to exhibit the mapper role.

### 3.4.4 Monitor

A robot's monitor role is to monitor the state of the mission (data acquisition and spraying tasks) and the area of interest to detect obstacles for safety and security reasons.

In the viticulture use case, a monitor robot in autonomous pest management plays a monitor role to observe the mission and environment.

The robot needs to carry visual sensors to perceive the environment and localization, navigation, monitor, and control components to exhibit a monitor role.

## 3.5 Robotic Behaviors

The level of autonomy of robotic systems dictates the humans required to operate them and the complexity and adaptability of the system. Full autonomy in real-world situations is currently challenging to apply in actual

use cases. However, there is a strong preference for semi-autonomous behaviors, rather than complete manual control, to reduce the attentional load on the operator or allow them to multitask or operate multiple systems simultaneously. However, it is considered essential to have humans in the loop to guide the robot's behaviors on tasks that typically evolve dynamically during the mission.

### 3.5.1 Autonomous

A robot's autonomous behavior is to make decisions (in response to external inputs) that are made within the robot system and do not involve human decision-making [67]. Autonomous robots can accomplish complex objectives on their own without humans. They use sensors to perceive the environment around them and then apply decision-making to take the optimal next step based on their data and mission. Autonomous robots are generally mobile and can therefore move around on their own. Like all other mobile robots, autonomous robots come in many different forms — from flying drones to ground-based robots to water-based and even underwater machines. However, autonomous robots must have some kind of monitoring and direct human control in case of an emergency.

In the viticulture use case, the robot needs to operate autonomously beyond the visual line of sight during operation to overcome any obstacles that might appear. The robot depends on perception, mapping, localization, navigation, and control components to exhibit autonomous behavior.

### 3.5.2 Semi-autonomous

A robot's semi-autonomous behavior is to make decisions (in response to external inputs) during operation, but a human is involved in some of this process and provides some direct control [67]. The human user is typically not involved in a semi-autonomous robot system unless necessary, unlike the shared system control between the computer and the human user in a semi-autonomous robot system.

In security management of MRS-based Assembly Lines use case, Gearbox (GB) infeed from upstream assembly line via workpiece carrier, Robot 1 takes GB from infeed and goes to sealant application. Afterward, drop GB on the Rotary Indexing Table (RIT) for operator workbench manual tasks. Upon completion, the operator leaves the safety zone, RIT clocks into the station, and robot 1 brings GB to RIT for secondary robots. The first position of RIT includes tightening of front panel bolt via double nutrunner, the second position includes stud setting by nutrunner. Infeed of studs from the auxiliary part feeder. After stud setting, Robot 1 takes GB from RIT and drops it on the second operator workbench RIT for manual tasks. When the finished operator leaves the safety area again for RIT to turn and Robot 1 to pick up a part and deliver to workpiece carrier on outfeed to the downstream machine.

The robot depends on an operator to perform manual tasks along with perception, mapping, localization, navigation, disinfection, and control components to exhibit semi-autonomous behavior.

### 3.5.3 Manual

A robot's manual behavior is to make decisions (in response to external inputs) during operation, but a human is involved in the complete process and provides direct control.

In the viticulture use case, the first set of UAVs used to monitor and collect high-quality multi-spectral images and 3D data from vineyards. This data will be analyzed, with minimal human interaction, to establish the physiology of individual vines and identify areas where fungicide should be applied, leading to the generation of an autonomous fungicide spraying mission for the robotic team.

# 4 Proposed Stratagem Concept and Methodology

This section describes the concept and methodology of *stratagem* with a sequence of steps on how a *robotics engineer* can select a robot according to the required capabilities, roles, and skills for an MRS mission-specific task. To this end, we use the stakeholders introduced in Section 4.1 to discuss how the stratagem concept support stakeholders to perform their tasks. We also discuss how stratagem concept and methodology works towards robotic or MRS mission configuration needs and how stakeholders play a role in them in Section 4.2 and 4.3.

## 4.1 Stakeholders

A stakeholder is an individual or organization having a right, share, claim, or interest in a system or in its possession of characteristics that meet their needs and expectations [68]. From the stratagem concept perspective, the main stakeholders in the SESAME project are as follows:

- **Robotic engineer**: The robotic engineer is responsible for specifying and configuring the hardware, software, and required capabilities using stratagem and closely interacts with the safety and security engineers.
- **Domain experts**: People with extensive knowledge of robotic-level and mission-level complex systems and experienced users of these systems.
- **Domain literature**: Domain literature contains information on multi-robot systems from articles, papers, thesis, and evaluations of various multi-robot systems.
- **Knowledge base**: A knowledge base entity contains reusable components with robust robotic-level and mission-level capabilities. These robust capabilities are derived from the domain experts or domain literature, which allow composing independent robotic missions and collaborative MRS missions.
- **Safety engineer**: The safety engineer is responsible for defining safety-related system aspects and closely interacts with the robotics engineer.
- **Security engineer**: The security engineer is responsible for defining security-related system aspects and closely interacts with the robotics engineer.

## 4.2 Stratagem Concept

An essential component of ExSce is the *stratagem* that encapsulates the robotic team's capabilities and individual robots. *Stratagem* concept enables novel and reusable capabilities that reinforce MRS robustness deployable across a large variety of different MRS missions. Models of these capabilities are incorporated in the SESAME knowledge base. They can be used for the specification of ExSce and transformation into executable models for simulation or the real MRS.

## 4.3 Stratagem Methodology

The stratagem enables engineers to specify and configure the hardware, software, and required capabilities. Using stratagems, engineers can select the robot and its onboard sensors based on robot capabilities, skills, roles, and behaviors by following Table 1 for the various use cases in the SESAME project.

The concept of stratagem methodology presented in Figure 5 has main steps that are performed sequentially and iteratively. This concept supports two distinct types of configurations. In the first case, the stratagem facilitates the configuration of robotic-level mission scenarios. In the second case, the stratagem supports the configuration of collaborative mission-level scenarios.

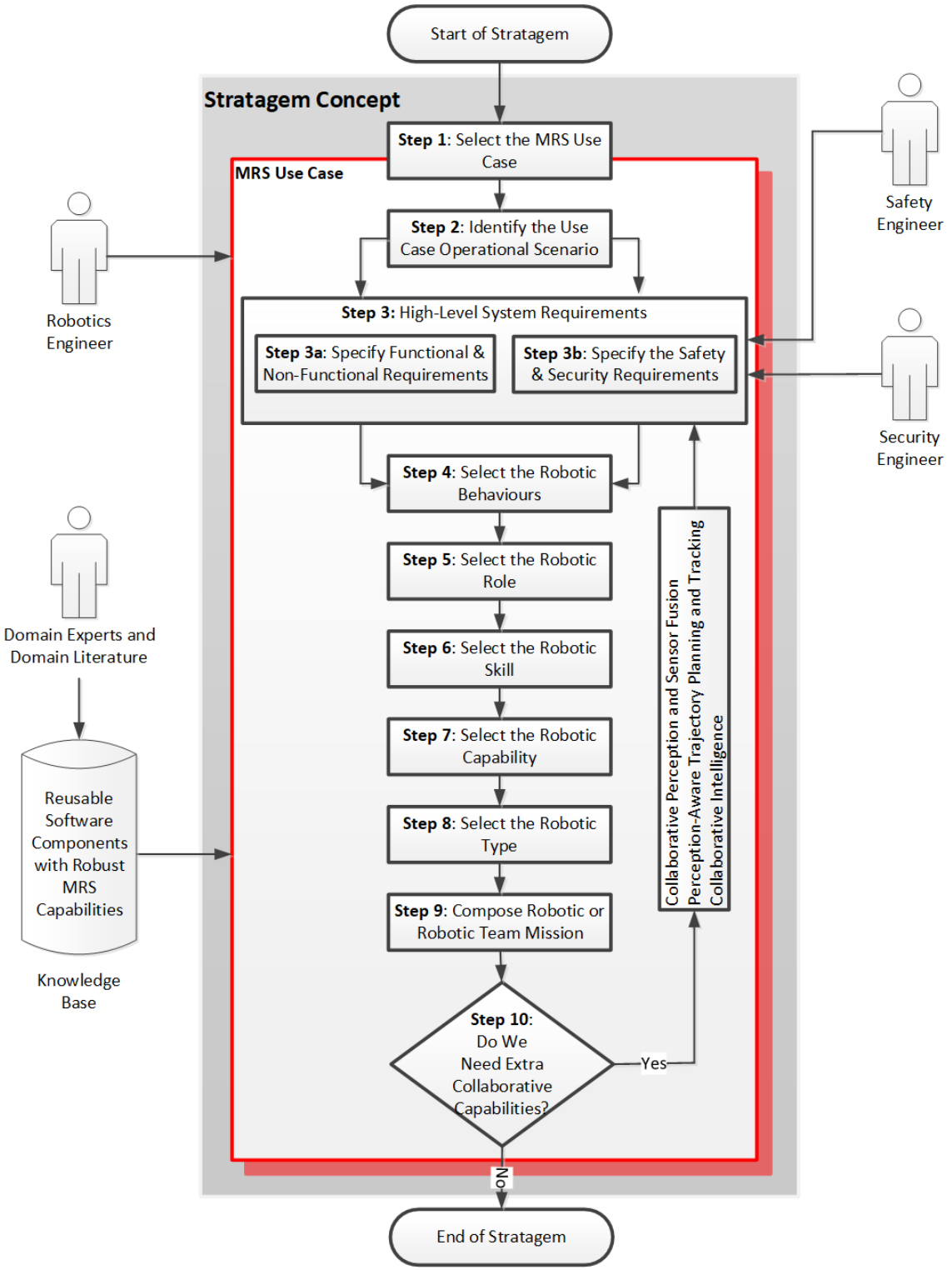


Figure 5: Stratagem Concept to Compose Missions or Collaborative MRS Missions.

### 4.3.1 Step 1: Select the MRS Use Case

The *robotics engineer* selects any one of the following MRS use cases to apply the stratagem concept in the SESAME project:

- Use Case 1: Dependable Multi-Robot Systems in Battery Innovation Centre
- Use Case 2: Disinfecting Hospital Environments using Robotic Teams
- Use Case 3: Power Station Inspection using Autonomous Multi-Robot Systems
- Use Case 4: Autonomous Pest Management in Viticulture
- Use Case 5: Security Management of MRS-Based Assembly Lines

### 4.3.2 Step 2: Identify the Use Case Operational Scenario

The *robotics engineer* identifies the operational context of the use case using the operational scenario for different modes of operation based on the *knowledge base*. A *knowledge base* entity contains reusable components with robust robotic-level and mission-level capabilities. These robust capabilities are derived from the *domain experts* or *domain literature*, which allow composing independent robotic missions and collaborative MRS missions.

### 4.3.3 Step 3: Specify the High-level System Requirements

The operational scenarios in terms of capabilities are transformed into high-level system requirements.

### 4.3.4 Step 3a: Specify the Functional Requirements and Non-Functional Requirements

The high-level system requirements are transformed into a set of functional, non-functional, safety, and security requirements. The functional requirements are transformed into a functional architecture identifying the functional hierarchy, functional behavior, and functional interfaces. The non-functional requirements are transformed into a set of quality attributes, for example, usability, reliability, performance, and supportability.

### 4.3.5 Step 3b: Specify the Security and Safety Requirements

The high-level system requirements are also transformed into functional safety and security requirements. The *safety engineer* is responsible for defining safety-related system aspects and closely interacts with the *robotics engineer*. The *security engineer* is responsible for defining security-related system aspects and closely interacts with the *robotics engineer*.

### 4.3.6 Step 4: Select the Robotic Behaviour

The *robotics engineer* selects the robotic behavior based on operational scenarios, high-level requirements, and *knowledge base* (see Section 3.5):

- Autonomous
- Semi-autonomous
- Manual

### 4.3.7 Step 5: Select the Robotic Role

The *robotics engineer* selects the robotic role based on operational scenarios, high-level requirements, behaviors, and *knowledge base* (see Section 3.4):

- Mapper
- Sprayer
- Monitor

### 4.3.8 Step 6: Select the Robotic Skill

The *robotics engineer* selects the robotic skill based on operational scenarios, high-level requirements, behaviors, robotic roles, capabilities (sensors/software components), and *knowledge base* (see Section 3.3):

- Mapping
- Tracking
- Manipulation
- Stacking
- Inspection
- Spraying
- Disinfection

### 4.3.9 Step 7: Select the Robotic Capability

The *robotics engineer* selects the robotic capability based on its operational scenario (see Section 3.2).

- Mobility (Fly or Ground Movement)
- Grasp
- Vision

### 4.3.10 Step 8: Select the Robotic Type

The *robotics engineer* selects the robotic type based on its operational scenario and capabilities:

- AGV
- UAV
- Robotic Arm

### 4.3.11 Step 9: Compose Individual Robot Mission or Robotic Team (MRS) Mission

The *robotics engineer* creates the robotic-level mission (Non-MRS) or robotic team-level mission (MRS) for the corresponding operational scenario using the above sequential steps.

### 4.3.12 Step 10: Do We Need Extra Collaborative Capabilities?

The *robotics engineer* checks for collaborative robotic team missions operational scenarios if any. If not, the robotics engineer ends the stratagem methodology. If yes, the robotics engineer repeats this process from step 3 to perform the collaborative MRS mission using team-level capabilities (see Section 3.1):

- Collaborative perception and sensor fusion
- Perception-aware trajectory planning and tracking
- Collaborative intelligence

# 5 A Case Study: Autonomous Pest Management in Viticulture

The following section describes the generic and application-specific architectural guidelines on composing and configuring the robotic-level and MRS mission-level capabilities into a more specific application that can instantiate for a diverse set of scenarios. We illustrate robotic-level and collaborative MRS mission-level capabilities with an autonomous pest management use case that involves sets of robots operating in a dynamic environment.

## 5.1 System Overview

In the context of SESAME, the MRS system in the autonomous pest management use case completes a mission by collaborating UAVs with different tasks [69], is as shown in Figure 6.

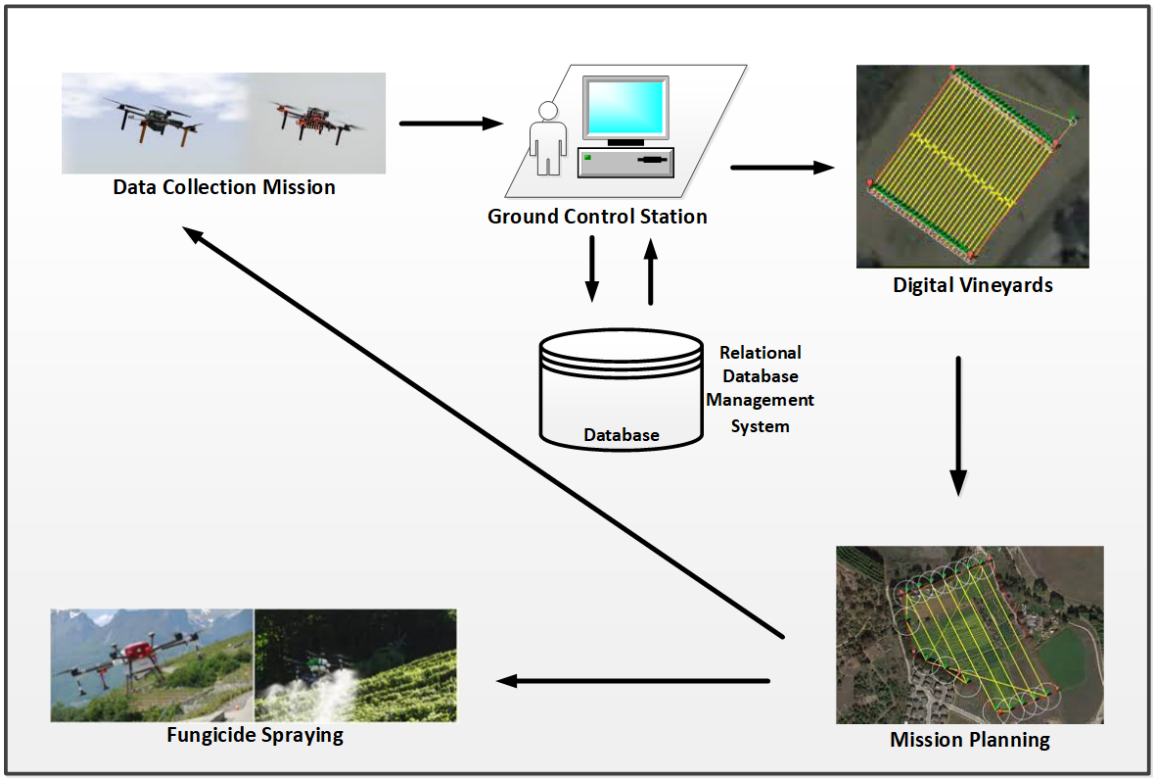


Figure 6: Autonomous Pest Management System Overview [69].

The MRS has sufficient processing power to execute collaborative MRS capabilities with multiple onboard sensors. The first set of UAVs is used to monitor and collect high-quality multi-spectral images and 3D data from DKOX's vineyards. With minimal human interaction, this data is analyzed to establish the physiology of individual vines and identify areas where fungicide should be applied, leading to the generation of an autonomous fungicide spraying mission for the robotic team. The autonomous fungicide spraying mission employs Aero41's AGv2 UAVs, equipped with its proprietary flight control system and pesticide spraying capabilities, and UAVs from the University of Luxembourg (UL), including customized versions of DJI Matrice 600 and DJI Matrice 210.

## 5.2 Research Goals

In order to perform autonomous pest management, the following research goals need to be achieved:

### 5.2.1 Goal G1: Operationalization of the Robot-based Data Analysis

An intelligent workflow has strived for data acquisition to operationalize the robot-based data analysis. For example, a map of potential risk areas is produced after a small-scale data collection over large areas. This map is used by specially equipped robots that autonomously perform a locally focussed in-depth analysis to detect specific stressors within the vineyards.

### 5.2.2 Goal G2: Optimal Fungicide Spraying Operation

Reduce fungicide spraying by 30%, contributing to environmental preservation. Optimal fungicide spraying will be achieved by combining LXSNS's algorithms for enhanced mapping and plant physiology identification at vine level with the novel MRS capabilities for sensor fusion and perception-aware trajectory planning.

### 5.2.3 Goal G3: Increase Productivity of Pest Management

Compared to using a tractor or a single UAV for pest management, increase productivity by 60% and 30%, respectively.

### 5.2.4 Goal G4: Extract and Use ExSce Templates in Vineyards

Using ExSce to model the fungicide spraying mission enables not only to capture environment and mission characteristics but also to extract and use ExSce templates in vineyards with different ground morphology but whose characteristics resemble the scenarios captured by those templates.

### 5.2.5 Goal G5: Reduce Operating Costs of Pest Management

Reduce operating costs by 40% by reducing human involvement to the absolute minimum.

### 5.2.6 Goal G6: Safety and Security Models

Using EDDI to instrument the UAV team with safety and security models enables the team to anticipate potential malfunctions or external attacks and initiate mitigation actions to bring the affected robot to a safe stop.

## 5.3 REQ 1.0: System Requirements

The use case partner needs to define the services that the system user expects in terms of system requirements, not how the system meets these requirements. The Requirements Diagram captures the use case partner needs in terms of system requirements such as functional, non-functional, safety, and security requirements, shown in Figure 7. A requirement diagram is a diagram to show requirements and the relationship among requirements. We specified categories (functional, non-functional, safety, and security) and identification numbers (1.0, 1.1, 1.2, 1.3) for each requirement. A satisfy relationship is a dependency between two requirements in a requirement diagram that can be derived from the high-level requirement.

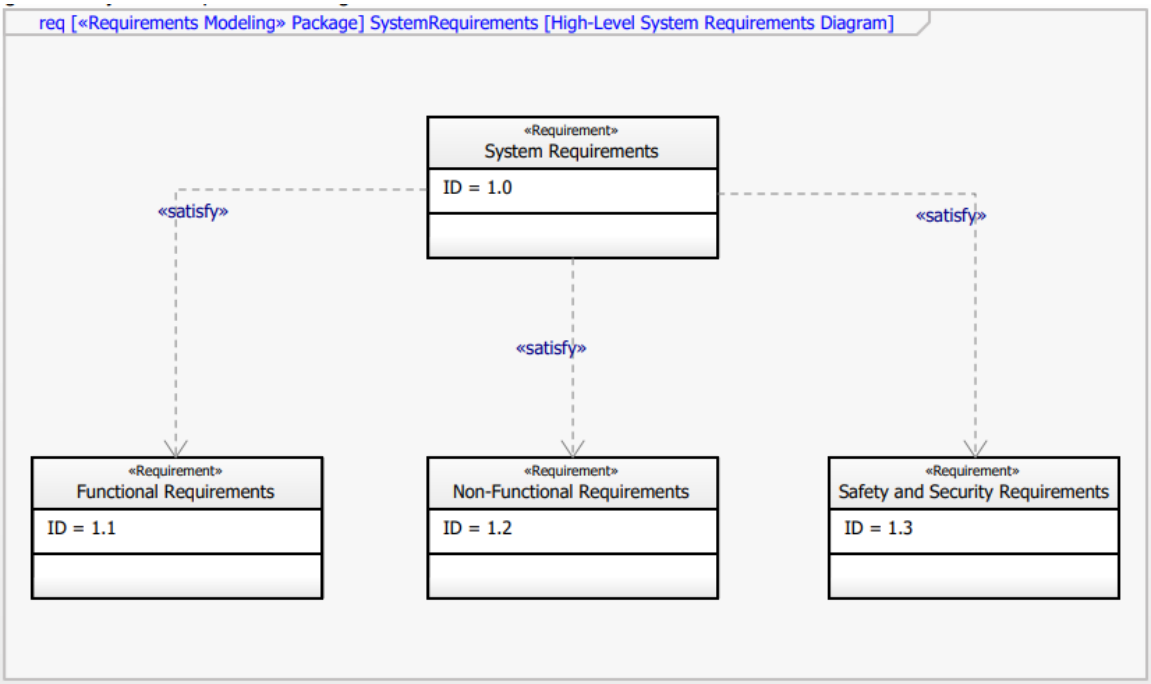


Figure 7: High-Level System Requirements Diagram of Autonomous Pest Management System.

### 5.3.1 REQ 1.1: Functional Requirements

The most important functional requirements of the autonomous pest management use case are (see Figure 8):

1. **REQ 1.1.1: Creation and Maintenance of Digital Vineyards**

   Based on available geodata datasets or robot-based data collection, an initial digital representation of the vineyards are set up. The included datasets include, among others, a digital elevation model and vine inventory. The vine inventory contains characteristics such as exact XY positioning, plant age, plant breed, and soil characteristics. During the vegetative period of the year, this digital vineyard will be completed by plant physiological characteristics and updated. The digital vineyard is set up and maintained in a relational database management system accessible through a Python-based interface.

2. **REQ 1.1.2: Surveillance**

   UAVs are deployed for real-time monitoring, surveillance, and coordination purposes for both missions (data collection and spraying). These UAVs should be able to take flight for at least the same time as the spraying drone to monitor the whole mission without being a bottleneck. In addition, it will carry visual and laser sensors to perceive the environment and evaluate the state of the mission.

3. **REQ 1.1.3: Data Collection**

   Different types of sensors need to be flown in the area of interest (AOI). Besides simple RGB cameras, highly specialized sensors such as high point density LiDAR scanners, 5-10 band multispectral cameras, and thermal cameras need to be carried by MRS. Every sensor type has its own flight parameter requirements such as altitude, inter-line distance, speed, adequate environmental conditions. Synchronized to UAV-based data collections, many side parameters need to be collected in order to derive, calibrate and validate various data products. These side parameters include temperature and humidity, spectroscopy of reference targets and incoming light, incoming light intensity, and cloud cover. This IoT-based setup needs to transmit the data packages autonomously to a cloud-based database.

4. **REQ 1.1.4: Data Processing and Disease or Stress Detection**

Raw data collected by MRS need to be transferred to a computational infrastructure to perform data pre-processing and data analysis. Data pre-processing includes orthorectification of FPA based camera systems and data fusion of IMU and GPS data streams. Once the data is processed up to calibrated continuous spatial data, vegetation analysis is performed using classification algorithms such as CNN, Support Vector Machines, and Random Forest classifiers. The goal consists of evaluating every vine in terms of plant health, physiology, and stress.

5. **REQ 1.1.5: Mission Planning**

   Autonomous mission planning holds the biggest potential in optimizing automated pest management in terms of efficiency and sustainability. A detailed digital vineyard allows the definition of an optimal mission: i) In-depth data collection for stress type discrimination requires complex sensors and time-consuming data analysis. A smart mission planning foresees only collecting data on spatial subsets of high probability infections, and ii) Once the infected vines are detected, the mission for fungicide spraying needs only to cover the subset of infected vines and not the whole vineyard.

6. **REQ 1.1.6: Collision Avoidance**

   MRS need to constantly monitor the environment and react autonomously when specific objects are detected: i) When detecting obstacles within the flight path, the robots should decide on an optimal object bypass, and ii) During a failsafe operation, the mission should abort and Return To Home (RTH) based on previously collected known points, free of obstacles, to quickly return to land.

7. **REQ 1.1.7: Fungicide Spraying**

   During fungicide spraying operation, the robot needs to navigate towards an infection hotspot and apply a predefined variable rate of fungicides. This rate should vary depending on the grade of infection. Therefore, high importance is set on the adequate amount of fungicide applied to a specific hotspot. To this end, the spraying UAV needs very precise positioning accuracy in order to maintain a low but constant distance to the vines. As the efficiency and security of the fungicide spraying rely on high positioning accuracy, a fall-back system should be implemented in case of loss of positioning accuracy, e.g. due to loss or RTK correction data.

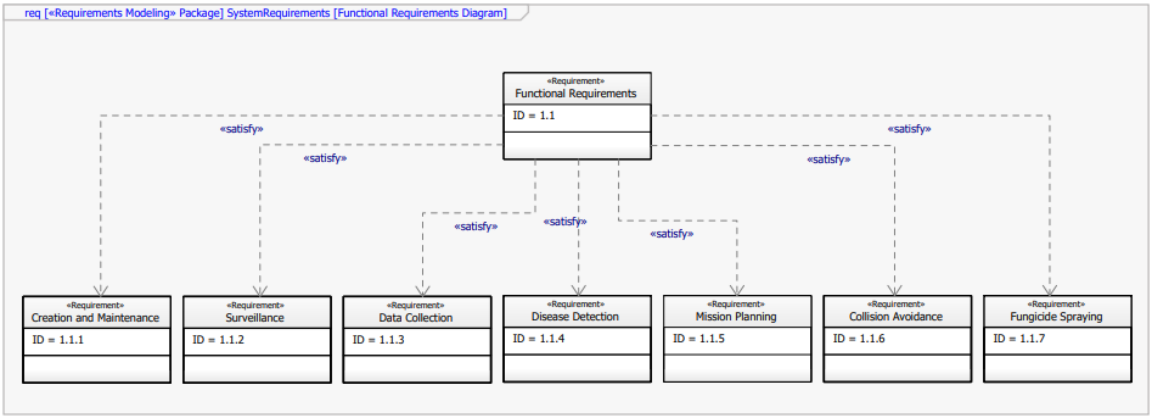


Figure 8: Functional Requirements Diagram of Autonomous Pest Management System.

### 5.3.2 REQ 1.2: Non-Functional Requirements

The most essential non-functional requirements of autonomous pest management are (see Figure 9):

1. **REQ 1.2.1: Usability**

   Autonomous robots need to coordinate different tasks within data collection, data processing, decision making, and vine management tasks. This should optimize the complete workflow for winemakers and require (none or) only minimal human interaction.

2. **REQ 1.2.2: Reliability**

   As the MRS are deployed in vineyards and not in a controlled environment, all the procedures need to be independent of varying meteorological conditions (e.g., significant changes in temperature and pressure, dust, precipitation). Although spraying is not permitted during high wind velocities or turbulence, increases in wind speed or turbulences must be considered during UAV roll-out. High wind velocities or turbulences have a high potential for security risks (e.g., UAV loss or wind drift when applying chemical compounds).

3. **REQ 1.2.3: Performance**

   As the ideal workflow foresees locally adapted vine management tasks executed by autonomous MRS based on an analysis of data collected by other autonomous MRS, high demands are set on the accuracy of the robot localization. A robot localization accuracy of 2-3 cm (XY) and 5 cm (Z) is required in a national or global coordinate reference system (CRS).

4. **REQ 1.2.4: Supportability**

   Precision agriculture, or more specifically precision viticulture is based on optimizing the complete agricultural system, increasing crop quality and crop yield while reducing the environmental footprint of agriculture. Emerging technologies, such as geospatial technologies, the Internet of Things (IoT), Big Data analysis, and artificial intelligence (AI), are considered to bring a huge potential in analyzing an agricultural system in all necessary detail. To this end, a lot of data is needed to describe the subject in the required level of detail. Remote sensing techniques allow the collection of data in a high temporal and spatial resolution and available shortly after data collection. The data collection can be understood as a multilayer process in which different sensors are used to collect the different types of needed data. To gather a more "general overview" of the area of interest (AOI) some sensors are foreseen to cover the complete AOI. In the case of a suspect hotspot (e.g., with a high risk of infection), other, more specific sensors are sent to these subsets in order to collect data for an in-depth analysis. All of these data acquisition tasks should/could be distributed on MRS in order to increase area/time efficiency, In addition, for safety and security reasons, all the data acquisition and spraying tasks are framed by "observing" robot systems. These robot systems monitor the MRS (e.g., position, speed, etc.) and the AOI in order to detect obstacles (e.g., people, trees, air traffic, etc.).

### 5.3.3 REQ 1.3: Safety and Security Requirements

The multi-robot systems foreseen in this use case bring a high potential of danger due to the fact of using mid-sized UAVs (5-50 kg) that can carry significant quantities of chemical compounds. In addition, the systems should operate autonomously beyond the visual line of sight so that no operator is able to intervene in case of an incident. Examples of hazardous situations are the failure of the UAV to detect a blocking obstacle (collision avoidance). This failure leads to undesired spraying (recognition of human beings or animals, loss of control due to a cyber-attack, UAV crossing the boundaries of the plot).

The safety and security of these systems need to be guaranteed for the following potential risks requirements (see Figure 10):

1. **REQ 1.3.1: Blocking Objects**

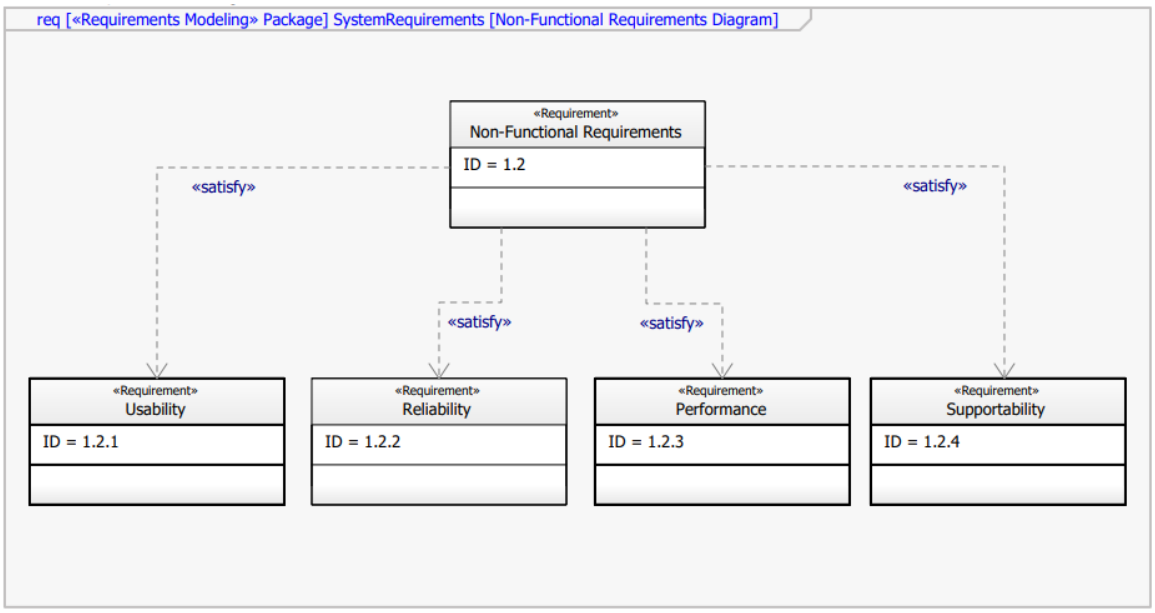


Figure 9: Non-Functional Requirements Diagram of Autonomous Pest Management System.

The MRS need to constantly monitor the environment to detect blocking objects, such as birds, humans, cars, and vegetation. If the MRS detects blocking objects within the flight path, the robots should decide on an optimal object bypass or navigate around the objects. If it is impossible to bypass or navigate around the objects, the robots should abort the mission and return home.

2. **REQ 1.3.2: Spray Chemical Compounds**

   As it is foreseen that the MRS shall spray chemical compounds, it needs to be guaranteed that none will get in contact with surfaces outside the area of interest.

3. **REQ 1.3.3: High Wind Velocities or Turbulences**

   As the MRS are deployed in vineyards and not in a controlled environment, all the procedures need to be independent of varying meteorological conditions (e.g., significant changes in temperature and pressure, dust, precipitation). Although spraying is not permitted during high wind velocities or turbulence, increases in wind speed or turbulences must be considered during UAV roll-out. High wind velocities or turbulences have a high potential for security risks (e.g., UAV loss or wind drift when applying chemical compounds).

## 5.4 Operational Scenarios

The above-explained research goals in Section 5.2 and system requirements in Section 5.3 are accomplished by fulfilling the various operational scenarios, which are explained below:

### 5.4.1 Scenario S1: Autonomous mission planning

In order to optimize efficiency and indirectly the operating costs of autonomous pest management by MRS, autonomous mission planning is required. To this end, a digital copy of the complete system of vineyards needs to be established and kept updated during the vegetative period of the year. This digital vineyard includes datasets such as a high-resolution 3D model of the vineyard, including digital 3D vine models and plant physiology.

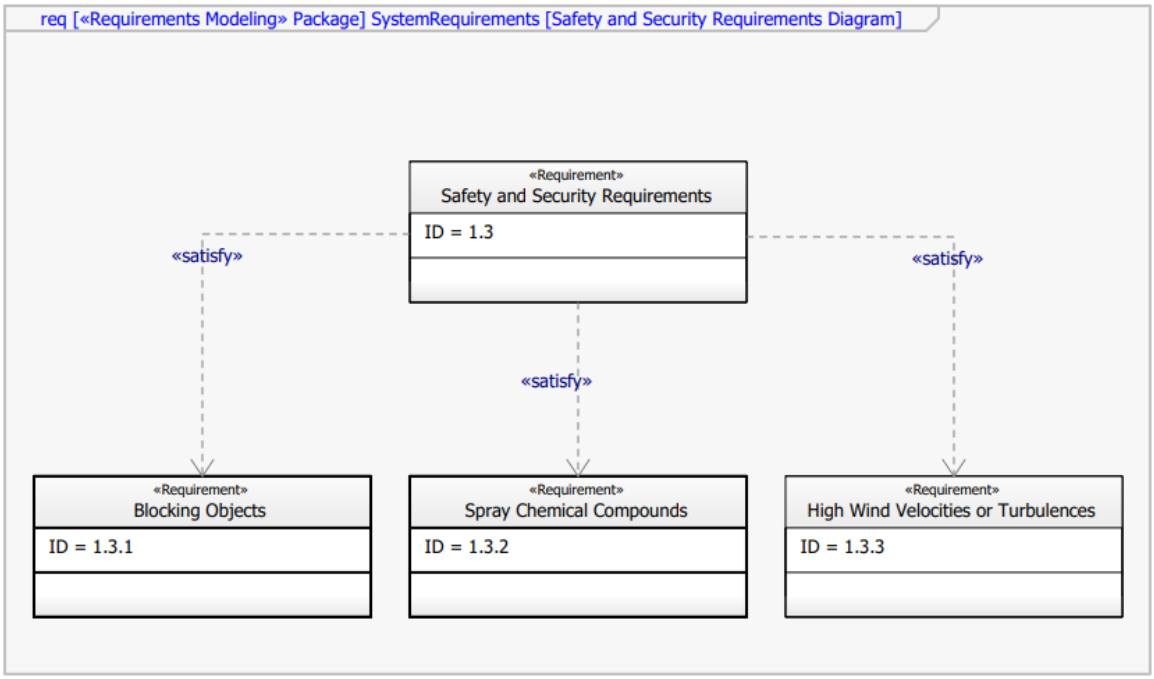


Figure 10: Safety and Security Requirements Diagram of Autonomous Pest Management System.

Figure 11 provides a graphical representation of a digital vineyard. Note that every single vine is localized based on its XYZ real-world coordinates [69]. The underlying relational database stores all the available and collected data and is subjected to classification algorithms that estimate probabilities for various parameters such as infection risk.

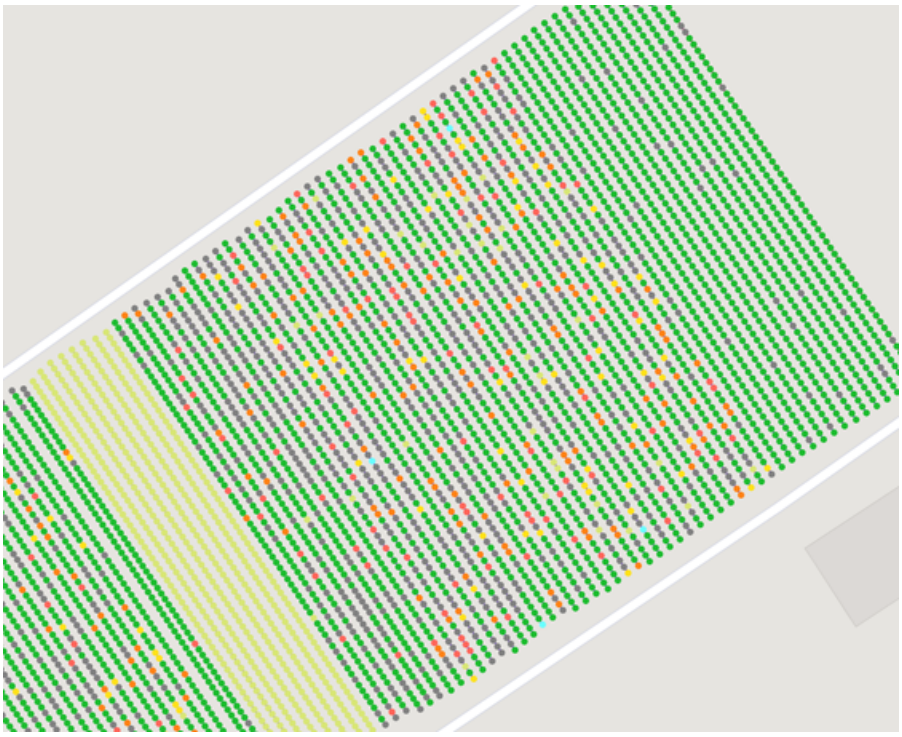

Figure 11: Graphic representation of a digital vineyard [69].

These high-resolution digital vineyards are used for smart autonomous planning. This operational scenario can be divided into the following categories:

1. **Mission planning for continuous data collection**: As the identification of a specific stressor requires advanced sensors and advanced data analysis procedures, these should only be applied on hotspots with a high probability of infection. To this end, data collection is run by a smart multiscale approach, limiting the use of complex sensors on hotspots detected by more simple sensor systems. These hotspots are planned to be derived from a summarizing data collection (on a tractor, spraying UAV or dedicated data collection UAV). An in-depth data collection spatially limited to previously detected hotspots should increase the efficiency of data collection and for fungicide spraying, too.
2. **Mission planning for fungicide spraying**: The efficiency of fungicide spraying depends among other parameters on the localization precision of the spraying task. The precision in XY being obvious, the precision in height is very important in order to achieve an optimal spraying result. To this end, a constant height of 2.5 meters above the canopy should be constantly kept which is considered during a smart mission planning considering the high-resolution terrain models and vine height data products. Similar to the in-depth data collection, spraying should also only be applied on vines with a very high probability of infection. Spraying tasks should considerably increase efficiency, when mission planning only focuses on previously detected hotspots.
3. **Mission planning in general**: As this use case plans MRS being fully exposed to environmental conditions, and time windows being highly variable due to meteorological conditions, the mission time should be planned to be as short as possible. In general, a smart mission planning should foresee running multiple robots simultaneously when large missions need to be completed. Splitting the area of interest in multiple subsets, each treated by a dedicated robot, allows reducing the mission duration significantly.

### 5.4.2 Scenario S2: Assessing security of surveillance

The system should operate autonomously beyond the visual line of sight so that no operator is able to intervene in case of an incident. In this scenario, safety tools are monitored and assessed in the vineyards with a focus on:

1. **Object detection**: Object detection within or in close proximity to the flight path, as well as in the area of fungicide spraying (person, birds, vehicles, etc.).
2. **Attack-proof**: Protection against potential external cyber-attacks.

### 5.4.3 Scenario S3: Assessing interaction and coordination of surveillance and spraying

The MRS foreseen in this use case brings a high potential of danger due to the fact of using mid-sized UAVs (5-50 kg) that can carry significant quantities of chemical compounds.

In this scenario, communication tools are monitored and assessed in the vineyards with a focus on:

1. **Object detection and collision avoidance**: Object detection and collision avoidance (person, birds, plant, etc.).
2. **Localisation accuracy**: Localisation accuracy (staying within the bounding box), interaction, and coordination between UAVs.

### 5.4.4 Scenario S4: Planning of fungicide spraying

Continuous data collection is expected to allow for the precise planning of fungicide spraying. This, in turn, will allow for a decrease of the average time to treat a target area. In addition, it is expected to reduce fungicide spraying contributing also to environment preservation.

Compared to using a tractor or a single UAV for pest management, one of the goals is to increase productivity substantially.

### 5.4.5 Scenario S5: Assessing fungicide spraying

In this scenario, we will assess fungicide application in vineyards with a focus towards:

1. **Precise treatment**: Precise treatment (depending on variable weather conditions), and
2. **Application quality**: Application quality (homogenous vs. variable rate application of fungicides in relation to the task to be executed).

## 5.5 Methodology Guidelines

The stratagem concept can be used as a generic methodology for engineers to specify and configure the hardware, software, and required robotic capabilities for various use cases. The main goal of using stratagem methodology with this use case is to create a set of generic and application-specific architectural guidelines on how to compose and configure the MRS capabilities for a use case application that can be instantiated for a diverse set of scenarios.

The stratagem concept and methodology presented in Section 4 has been applied to an autonomous pest management use case to perform robotic missions and collaborative MRS missions such as data collection, mission planning, surveillance, and spraying operations (see Table 2).

Table 2: Scenarios were derived from autonomous pest management use case.

| ID | Scenarios | Description |
|---|---|---|
| Scenario 1 | Autonomous mission planning | Continuous data collection |
| Scenario 2 | Assessing surveillance security | Safety and security are monitored |
| Scenario 3 | Assessing coordination interaction | Communication tools are monitored |
| Scenario 4 | Planning of fungicide spraying | Spraying fungicide in vineyard |
| Scenario 5 | Assessing fungicide spraying | Fungicide operation is monitored |

This methodology supports two distinct types of configurations. In the first case, the stratagem facilitates the configuration of robotic mission scenarios (data collection and fungicide spraying) (see Section 5.6 for more details). In the second case, the stratagem configures collaborative MRS mission scenarios by reusing existing missions scenarios (data collection and fungicide spraying) combined and sequenced in new ways (monitor the state of the mission) (see Section 5.7 for more details).

## 5.6 Compose Robotic Mission

Let us assume that robotic mission-level operational scenarios are required to perform the data collection mission and fungicide spraying mission in an autonomous pest management use case. Figure 12 provides a graphical representation of an autonomous mission planning for continuous data collection and fungicide spraying operation in a digital vineyard (see Section 5.4).

Based on the selected operational scenario, we need two sets of UAVs to perform this mission-level operation. One set of UAVs are equipped with visual and laser sensors to data collection. The second set of UAVs are equipped with an adequate amount of fungicide spray and required onboard sensors to navigate towards an infection hotspot.


Figure 12: Scenario: Autonomous mission planning for continuous data collection and fungicide spraying. UAV 1: Data collection, and UAV 2: Fungicide spraying.

### 5.6.1 First Iteration: Compose Data Collection Mission

#### Step 1: Select the MRS Use Case

In this step, the robotics engineer selects the autonomous pest management in the viticulture MRS use case to apply the stratagem concept in the SESAME project:

- Use Case 1: Dependable Multi-Robot Systems in Battery Innovation Centre
- Use Case 2: Disinfecting Hospital Environments using Robotic Teams
- Use Case 3: Power Station Inspection using Autonomous Multi-Robot Systems
- **Use Case 4: Autonomous Pest Management in Viticulture**
- Use Case 5: Security Management of MRS-Based Assembly Lines

#### Step 2: Identify the Use Case Operational Scenario

In this step, the robotics engineer identifies the operational scenarios for different modes of operation based on the knowledge base (in this step, data collection), are as shown below:

- **Scenario 1: Autonomous mission planning for continuous data collection and fungicide spraying**
- Scenario 2: Assessing security of surveillance
- Scenario 3: Assessing interaction and coordination of surveillance and spraying
- Scenario 4: Planning of fungicide spraying
- Scenario 5: Assessing fungicide spraying

#### Step 3: Specify the High-level System Requirements

In this step, the operational needs are transformed into high-level system requirements. These high-level requirements are captured in Section 5.3. Based on the selected operational scenario, the high-level system requirements are transformed into a set of the following functional requirements:

- **Data collection**
- Fungicide spraying
- Monitor mission

The non-functional requirements are:

- **Usability**
- **Reliability**
- **Performance**
- **Supportability**

The safety and security requirements are:

- **Safety**
- **Security**

The safety engineer is responsible for defining safety-related system aspects and closely interacts with the robotics engineer. The security engineer is responsible for defining security-related system aspects and closely interacts with the robotics engineer.

**Step 4: Select the Robotic Behaviour**

In this step, the robotics engineer selects the robotic behavior based on the operational scenario, high-level requirements, and knowledge base (see Section 3.5):

- **Autonomous**
- Semi-autonomous
- Manual

In Table 1, we already introduced an evaluation matrix to choose an engineer for various robotic capabilities, skills, roles, behaviours, and components. The engineer selects the robotic behavior based on the use case operational scenario. In this step, the robotics engineer selects the set of robots with the highest mapping skill. In this operational scenario, the engineer selects the UAVs with autonomous behavior because the robots are required to fly and operate autonomously beyond the visual line of sight.

**Step 5: Select the Robotic Role**

In this step, the robotics engineer selects the robot role as a mapper because this robotic role is required in this operational scenario (see Section 3.4):

- **Mapper**
- Sprayer
- Monitor

**Step 6: Select the Robotic Skill**

In this step, the engineer selects the robot skill based on its operational scenario and capability evaluation matrix, as explained in Section 3.3: The robot must have the skill of mapping for map generation.

- **Mapping**
- Tracking
- Manipulation
- Stacking
- Inspection
- Spraying
- Disinfection

**Step 7: Select the Robotic Capability**

In this step, the robotics engineer selects the robotic capability based on its operational scenario (see Section 3.2).

- **Mobility** (**Fly**/Ground Movement)
- Grasp
- **Vision**

If we need a UAV to map generation in a vineyard, the robot must have the flying robotic capability. If we need an RGB camera to generate a vineyard map, the robot must have vision capability using an RGB camera.

**Step 8: Select the Robotic Type**

In this step, the robotics engineer selects the robotic type based on its operational scenario and capabilities:

- AGV
- **UAV**
- Robotic Arm

The robot has to cover a large vineyard to expedite the mapping process in a limited time. The 3D mobility over 2D will expedite the process of generating a map for a vineyard. Consequently, the robot should have 3D mobility, i.e., the robot should have a flying capability. Hence, the robotics engineer selects a UAV to generate a map vineyard.

**Step 9: Compose Individual Robot Mission or Robotic Team (MRS) Mission**

The *robotics engineer* creates the robotic-level mission (Non-MRS) or robotic team-level mission (MRS) for the corresponding operational scenario using the above sequential steps. In this step, the *robotics engineer* has created the following robotic-level missions:

- **Data collection**
- Fungicide spraying
- Monitor mission

**Step 10: Do We Need Extra Collaborative Capabilities?**

The *robotics engineer* checks for collaborative robot or robotic team missions operational scenarios if any. If yes, the robotics engineer repeats this process from step 3 to perform the collaborative MRS mission using mission-level capabilities (see Section 3.1):

- Collaborative perception and sensor fusion
- Perception-aware trajectory planning and tracking
- Collaborative intelligence

If not collaborative mission, the robotics engineer ends the stratagem methodology. Here, we consider the individual robotic mission. Hence, go to second iteration to perform another individual robotic mission (from Step 1).

### 5.6.2 Second Iteration: Compose Data Collection Mission + Fungicide Spraying Mission

**Step 1: Select the MRS Use Case**

In this step, the robotics engineer selects the autonomous pest management in the viticulture MRS use case to apply the stratagem concept in the SESAME project:

- Use Case 1: Dependable Multi-Robot Systems in Battery Innovation Centre
- Use Case 2: Disinfecting Hospital Environments using Robotic Teams
- Use Case 3: Power Station Inspection using Autonomous Multi-Robot Systems
- **Use Case 4: Autonomous Pest Management in Viticulture**
- Use Case 5: Security Management of MRS-Based Assembly Lines

**Step 2: Identify the Use Case Operational Scenario**

In this step, the robotics engineer identifies the operational scenarios for different modes of operation based on the knowledge base (in this step, fungicide spraying in addition to the previous scenario), are as shown below:

- **Scenario 1: Autonomous mission planning for continuous data collection and fungicide spraying**
- Scenario 2: Assessing security of surveillance
- Scenario 3: Assessing interaction and coordination of surveillance and spraying
- **Scenario 4: Planning of fungicide spraying**
- Scenario 5: Assessing fungicide spraying

**Step 3: Specify the High-level System Requirements**

In this step, the operational needs are transformed into high-level system requirements. These high-level requirements are captured in Section 5.3. Based on the selected operational scenario, the high-level system requirements are transformed into a set of the following functional requirements:

- **Data collection**
- **Fungicide spraying**
- Monitor mission

The non-functional requirements are:

- **Usability**
- **Reliability**
- **Performance**
- **Supportability**

The safety and security requirements are:

- **Safety**
- **Security**

The safety engineer is responsible for defining safety-related system aspects and closely interacts with the robotics engineer. The security engineer is responsible for defining security-related system aspects and closely interacts with the robotics engineer.

**Step 4: Select the Robotic Behaviour**

In this step, the robotics engineer selects the robotic behavior based on the operational scenario, high-level requirements, and knowledge base (see Section 3.5):

- **Autonomous**
- Semi-autonomous
- Manual

The engineer selects the robotic behavior based on the use case operational scenario. In this step, the robotics engineer selects the robots with the spraying skill. In this operational scenario, the engineer selects the UAVs

with autonomous behavior because the robots are required to fly and operate autonomously beyond the visual line of sight and perform fungicide spraying.

**Step 5: Select the Robotic Role**

In this step, the robotics engineer selects the robot role as sprayer because these robotic roles are required in this operational scenario (see Section 3.4):

- **Mapper**
- **Sprayer**
- Monitor

**Step 6: Select the Robotic Skill**

In this step, the engineer selects the robot skill based on its operational scenario and capability evaluation matrix, as explained in Section 3.3: The robot must have the skill of spraying for fungicide spraying operation.

- **Mapping**
- Tracking
- Manipulation
- Stacking
- Inspection
- **Spraying**
- Disinfection

**Step 7: Select the Robotic Capability**

In this step, the robotics engineer selects the robotic capability based on its operational scenario (see Section 3.2).

- **Mobility** (**Fly**/Ground Movement)
- Grasp
- **Vision**

If we need a UAV to spray fungicide in a vineyard, the robot must have the flying robotic capability.

**Step 8: Select the Robotic Type**

In this step, the robotics engineer selects the robotic type based on its operational scenario and capabilities:

- AGV
- **UAV**
- Robotic Arm

The robot has to cover a large vineyard to expedite the spraying operation in a limited time. Consequently, the robot should have 3D mobility, i.e., the robot should have a flying capability. Hence, the robotics engineer selects a UAV to perform a fungicide spraying task for a vineyard.

**Step 9: Compose Individual Robot Mission or Robotic Team (MRS) Mission**

The *robotics engineer* creates the robotic-level mission (Non-MRS) or robotic team-level mission (MRS) for the corresponding operational scenario using the above sequential steps. In this step, the *robotics engineer* has created the following robotic-level missions:

- **Data collection**
- **Fungicide spraying**
- Monitor mission

**Step 10: Do We Need Extra Collaborative Capabilities?**

The *robotics engineer* checks for collaborative robot or robotic team missions operational scenarios if any. If yes, the robotics engineer repeats this process from step 3 to perform the collaborative MRS mission using mission-level capabilities (see Section 3.1):

- Collaborative perception and sensor fusion
- Perception-aware trajectory planning and tracking
- Collaborative intelligence

Here, we consider the collaborative MRS mission. Hence, go to third iteration to perform collaborative MRS mission (from Step 3). If not collaborative mission, the robotics engineer ends the stratagem methodology.

### 5.6.3 Third Iteration: Compose Monitor MRS Mission to Monitor the Data Collection Mission + Fungicide Spraying Mission State

**Step 3: Specify the High-level System Requirements**

In this step, the operational needs are transformed into high-level system requirements. These high-level requirements are captured in Section 5.3. Based on the selected operational scenario, the high-level system requirements are transformed into a set of the following functional requirements:

- **Data collection**
- **Fungicide spraying**
- **Monitor mission**

The non-functional requirements are:

- **Usability**
- **Reliability**
- **Performance**
- **Supportability**

The safety and security requirements are:

- **Safety**
- **Security**

The safety engineer is responsible for defining safety-related system aspects and closely interacts with the robotics engineer. The security engineer is responsible for defining security-related system aspects and closely interacts with the robotics engineer.

**Step 4: Select the Robotic Behaviour**

In this step, the robotics engineer selects the robotic behavior based on the operational scenario, high-level requirements, and knowledge base (see Section 3.5):

- **Autonomous**
- Semi-autonomous
- Manual

In this operational scenario, the engineer selects the UAVs with autonomous behavior because the robots are required to fly and operate autonomously beyond the visual line of sight and monitor the state of the mission.

**Step 5: Select the Robotic Role**

In this step, the robotics engineer selects the robot role as monitor because these robotic roles are required in this operational scenario (see Section 3.4):

- **Mapper**
- **Sprayer**
- **Monitor**

**Step 6: Select the Robotic Skill**

In this step, the engineer selects the robot skill based on its operational scenario and capability evaluation matrix, as explained in Section 3.3: The robot must have the skill of tracking and inspection for monitoring the mission state.

- **Mapping**
- **Tracking**
- Manipulation
- Stacking
- **Inspection**
- **Spraying**
- Disinfection

**Step 7: Select the Robotic Capability**

In this step, the robotics engineer selects the robotic capability based on its operational scenario (see Section 3.2).

- **Mobility** (**Fly**/Ground Movement)
- Grasp
- **Vision**

If we need a UAV to monitor in a vineyard, the robot must have the flying robotic capability.

**Step 8: Select the Robotic Type**

In this step, the robotics engineer selects the robotic type based on its operational scenario and capabilities:

- AGV
- **UAV**
- Robotic Arm

The robot has to cover a large vineyard to expedite the monitoring operation in a limited time. Consequently, the robot should have 3D mobility, i.e., the robot should have a flying capability. Hence, the robotics engineer selects a UAV to perform a monitoring task for a vineyard.

**Step 9: Compose Individual Robot Mission or Robotic Team (MRS) Mission**

The *robotics engineer* creates the robotic-level mission (Non-MRS) or robotic team-level mission (MRS) for the corresponding operational scenario using the above sequential steps. In this step, the *robotics engineer* has created the following robotic-level missions:

- **Data collection**
- **Fungicide spraying**
- **Monitor mission**

**Step 10: Do We Need Extra Collaborative Capabilities?**

The *robotics engineer* checks for collaborative robot or robotic team missions operational scenarios if any. If yes, the robotics engineer repeats this process from step 3 to perform the collaborative MRS mission using mission-level capabilities (see Section 3.1):

- Collaborative perception and sensor fusion
- Perception-aware trajectory planning and tracking
- Collaborative intelligence

Similarly, the robotics engineer creates the robotic or collaborative MRS missions for all the operational scenarios. Here, we consider the end of the missions. Hence, the robotics engineer ends the stratagem methodology. See Section 5.7 for complete collaborative mission scenarios.

## 5.7 Compose Collaborative MRS Mission

Let us assume that a collaborative MRS mission-level operational scenario is required to monitor the whole data collection mission and fungicide spraying mission in an autonomous pest management. Figure 13 depicts an MRS mission scenario in autonomous pest management in which multiple UAVs are deployed to collect the data, spray the fungicide, and monitor the whole mission.

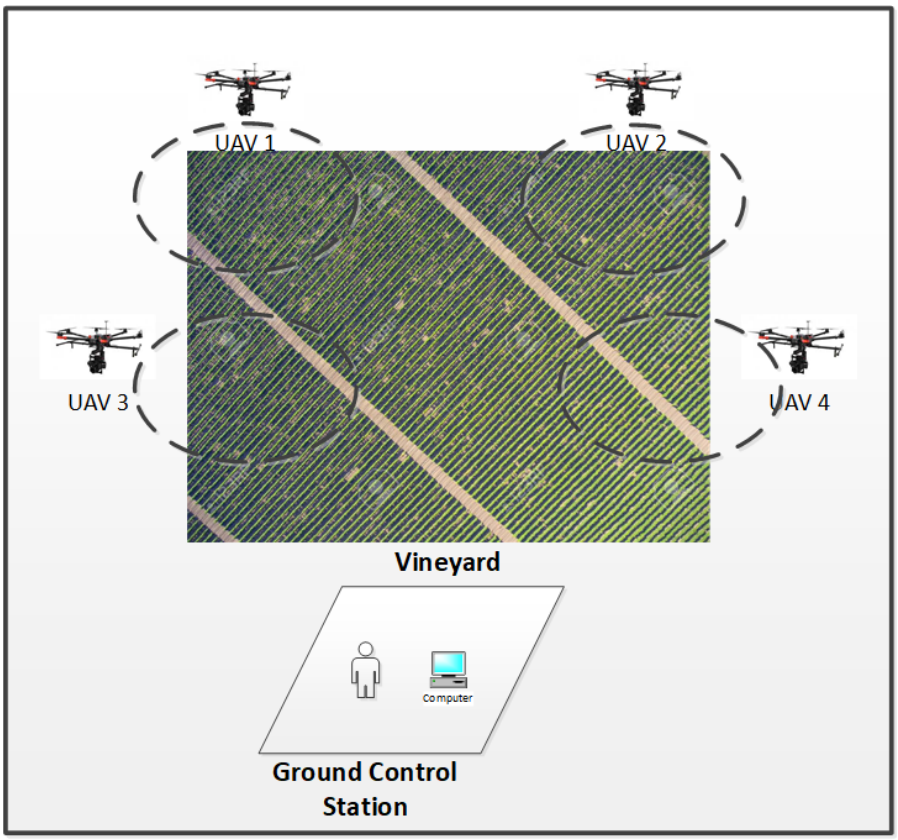


Figure 13: Compose Collaborative MRS Mission: Autonomous pest management use case.

### 5.7.1 Mission Description

UAV MRS mission is deployed for data collection, fungicide spraying, monitoring the whole mission, and coordination purpose in autonomous pest management use cases.

### 5.7.2 Pre-Conditions

One set of UAVs are equipped with visual and laser sensors to perceive the environment and evaluate the mission state. The second set of UAVs are equipped with an adequate amount of fungicide spray and required onboard sensors to navigate towards an infection hotspot.

### 5.7.3 Main Success Mission

1. UAVs take off and commence the continuous data collection (see Section 5.6, first iteration) and fungicide spraying (see Section 5.6, second iteration). These UAVs should fly at least the same time as the spraying drone to monitor the whole mission without being a bottleneck (see Section 5.6, third iteration).

2. The UAVs track and send the location and state of each UAV to the Ground Control Station.

3. The UAVs split the area of interest in multiple subsets, each treated by a dedicated robot, significantly reducing the mission duration.

4. One set of UAVs processes the digital vineyard includes datasets such as a high-resolution 3D model of the vineyard, including digital 3D vine models, and plant physiology using visual or laser sensors.

5. The Second set of UAVs detects the hotspots and applies fungicide spraying on vines with a very high probability of infection.

### 5.7.4 Alternate Mission

If any UAV has not had sufficient battery or loss of communication or adequate amount of fungicide spraying to continue the mission, it enables a safe "Return To Home" (RTH) to quickly return to land.

# 6 Conclusions

This document outlines the capabilities of MRS and architectural guidelines, thus setting the foundation for Executable Scenarios (ExSce) to be specified in WP3.

These novel and reusable capabilities reinforce the multi-robot system's robustness deployable across many different multi-robot systems use case missions such as:

- Use Case 1: Dependable Multi-Robot Systems in the Battery Innovation Centre.
- Use Case 2: Disinfecting Hospital Environments using Robotic Teams.
- Use Case 3: Power Station Inspection using Autonomous Multi-Robot Systems.
- Use Case 4: Autonomous Pest Management in Viticulture.
- Use Case 5: Security Management of Multi-Robot Systems-Based Assembly Lines.

These capabilities are incorporated in the SESAME knowledge base. They can specify ExSce and transform them into executable models for simulation or the real multi-robot systems.

This document will be updated throughout the duration of the project to address any issues that might arise.

# References


[1] SESAME, "Sesame: Secure and safe multi-robot systems, annex 1 - description of action," 2020. Last accessed 16 December 2021.

[2] K. Dantu, B. Kate, J. Waterman, P. Bailis, and M. Welsh, "Programming micro-aerial vehicle swarms with karma," in *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, pp. 121–134, 2011.

[3] L. Mottola, M. Moretta, K. Whitehouse, and C. Ghezzi, "Team-level programming of drone sensor networks," in *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, pp. 177–190, 2014.

[4] D. Dedousis and V. Kalogeraki, "A framework for programming a swarm of uavs," in *Proceedings of the 11th PErvasive Technologies Related to Assistive Environments Conference*, pp. 5–12, 2018.

[5] C. Pinciroli and G. Beltrame, "Buzz: An extensible programming language for heterogeneous swarm robotics," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3794–3800, IEEE, 2016.

[6] D. C. Silva, P. H. Abreu, L. P. Reis, and E. Oliveira, "Development of a flexible language for mission description for multi-robot missions," *Information Sciences*, vol. 288, pp. 27–44, 2014.

[7] S. García, P. Pelliccione, C. Menghi, T. Berger, and T. Bures, "High-level mission specification for multiple robots," in *Proceedings of the 12th ACM SIGPLAN International Conference on Software Language Engineering*, pp. 127–140, 2019.

[8] B. Schwartz, L. Nägele, A. Angerer, and B. A. MacDonald, "Towards a graphical language for quadrotor missions," *arXiv preprint arXiv:1412.1961*, 2014.

[9] D. Harel and M. Politi, *Modeling reactive systems with statecharts: the STATEMATE approach*. McGraw-Hill, Inc., 1998.

[10] J. Bohren and S. Cousins, "The smach high-level executive [ros news]," *IEEE Robotics & Automation Magazine*, vol. 17, no. 4, pp. 18–20, 2010.

[11] M. Klotzbücher and H. Bruyninckx, "Coordinating robotic tasks and systems with rfsm statecharts," *Journal of Software Engineering for Robotics*, vol. 3, no. 1, pp. 28–56, 2012.

[12] U. Thomas, G. Hirzinger, B. Rumpe, C. Schulze, and A. Wortmann, "A new skill based robot programming language using uml/p statecharts," in *2013 IEEE International Conference on Robotics and Automation*, pp. 461–466, IEEE, 2013.

[13] J. L. Peterson, "Petri nets," *ACM Computing Surveys (CSUR)*, vol. 9, no. 3, pp. 223–252, 1977.

[14] F.-Y. Wang, K. J. Kyriakopoulos, A. Tsolkas, and G. N. Saridis, "A petri-net coordination model for an intelligent mobile robot," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, no. 4, pp. 777–789, 1991.

[15] V. A. Ziparo, L. Iocchi, D. Nardi, P. F. Palamara, and H. Costelha, "Petri net plans: a formal model for representation and execution of multi-robot plans," in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*, pp. 79–86, 2008.

[16] D. Bozhinoski, D. Di Ruscio, I. Malavolta, P. Pelliccione, and M. Tivoli, "Flyaq: Enabling non-expert users to specify and generate missions of autonomous multicopters," in *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pp. 801–806, IEEE, 2015.

[17] J. A. Besada, L. Bergesio, I. Campaña, D. Vaquero-Melchor, J. López-Araquistain, A. M. Bernardos, and J. R. Casar, "Drone mission definition and implementation for automated infrastructure inspection using airborne sensors," *Sensors*, vol. 18, no. 4, p. 1170, 2018.

[18] A. P. Lamping, J. N. Ouwerkerk, and K. Cohen, "Multi-uav control and supervision with ros," in *2018 Aviation Technology, Integration, and Operations Conference*, p. 4245, 2018.

[19] S. Dhouib, S. Kchir, S. Stinckwich, T. Ziadi, and M. Ziane, "Robotml, a domain-specific language to design, simulate and deploy robotic applications," in *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*, pp. 149–160, Springer, 2012.

[20] G. E. Fainekos, A. Girard, H. Kress-Gazit, and G. J. Pappas, "Temporal logic motion planning for dynamic robots," *Automatica*, vol. 45, no. 2, pp. 343–352, 2009.

[21] C. Finucane, G. Jing, and H. Kress-Gazit, "Ltlmop: Experimenting with language, temporal logic and robot control," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1988–1993, IEEE, 2010.

[22] M. Guo and D. V. Dimarogonas, "Multi-agent plan reconfiguration under local ltl specifications," *The International Journal of Robotics Research*, vol. 34, no. 2, pp. 218–235, 2015.

[23] M. Guo, K. H. Johansson, and D. V. Dimarogonas, "Revising motion planning under linear temporal logic specifications in partially known workspaces," in *2013 IEEE International Conference on Robotics and Automation*, pp. 5025–5032, IEEE, 2013.

[24] H. Kress-Gazit, "Robot challenges: Toward development of verication and synthesis techniques [errata]," *IEEE Robotics & Automation Magazine*, vol. 18, no. 4, pp. 108–109, 2011.

[25] S. Maoz and J. O. Ringert, "Gr (1) synthesis for ltl specification patterns," in *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, pp. 96–106, 2015.

[26] C. Menghi, S. Garcia, P. Pelliccione, and J. Tumova, "Multi-robot ltl planning under uncertainty," in *International Symposium on Formal Methods*, pp. 399–417, Springer, 2018.

[27] A. Ulusoy, S. L. Smith, X. C. Ding, C. Belta, and D. Rus, "Optimal multi-robot path planning with temporal logic constraints," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3087–3092, IEEE, 2011.

[28] E. M. Wolff, U. Topcu, and R. M. Murray, "Automaton-guided controller synthesis for nonlinear systems with temporal logic," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4332–4339, IEEE, 2013.

[29] P. Doherty and J. Kvarnström, "Temporal action logics," *Foundations of Artificial Intelligence*, vol. 3, pp. 709–757, 2008.

[30] M. P. Ashley-Rollman, P. Lee, S. C. Goldstein, P. Pillai, and J. D. Campbell, "A language for large ensembles of independently executing nodes," in *International Conference on Logic Programming*, pp. 265–280, Springer, 2009.

[31] C. Menghi, C. Tsigkanos, T. Berger, P. Pelliccione, and C. Ghezzi, "Property specification patterns for robotic missions," in *Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings*, pp. 434–435, 2018.

[32] L. Kunze, T. Roehm, and M. Beetz, "Towards semantic robot description languages," in *2011 IEEE International Conference on Robotics and Automation*, pp. 5589–5595, IEEE, 2011.

[33] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson, "Feature-oriented domain analysis (foda) feasibility study," tech. rep., Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst, 1990.

[34] S. Gerasimou, N. Matragkas, and R. Calinescu, "Towards systematic engineering of collaborative heterogeneous robotic systems," in *2019 IEEE/ACM 2nd International Workshop on Robotics Software Engineering (RoSE)*, pp. 25–28, IEEE, 2019.

[35] J. Harbin, S. Gerasimou, N. Matragkas, A. Zolotas, and R. Calinescu, "Model-driven simulation-based analysis for multi-robot systems," in *2021 ACM/IEEE 24th International Conference on Model Driven Engineering Languages and Systems (MODELS)*, pp. 331–341, IEEE, 2021.

[36] N. Hochgeschwender, L. Gherardi, A. Shakhirmardanov, G. K. Kraetzschmar, D. Brugali, and H. Bruyninckx, "A model-based approach to software deployment in robotics," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3907–3914, IEEE, 2013.

[37] H. Bruyninckx, M. Klotzbücher, N. Hochgeschwender, G. Kraetzschmar, L. Gherardi, and D. Brugali, "The brics component model: a model-based development paradigm for complex robotics software systems," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pp. 1758–1764, 2013.

[38] N. Hochgeschwender, S. Schneider, H. Voos, H. Bruyninckx, and G. K. Kraetzschmar, "Graph-based software knowledge: Storage and semantic querying of domain models for run-time adaptation," in *2016 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, pp. 83–90, IEEE, 2016.

[39] A. W. Stroupe and T. Balch, "Mission-relevant collaborative observation and localization," in *Multi-Robot Systems: From Swarms to Intelligent Automata*, pp. 31–40, Springer, 2002.

[40] S. Dong, K. Xu, Q. Zhou, A. Tagliasacchi, S. Xin, M. Nießner, and B. Chen, "Multi-robot collaborative dense scene reconstruction," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–16, 2019.

[41] B. D. Gouveia, D. Portugal, D. C. Silva, and L. Marques, "Computation sharing in distributed robotic systems: A case study on slam," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 410–422, 2014.

[42] M. Vasic and A. Martinoli, "A collaborative sensor fusion algorithm for multi-object tracking using a gaussian mixture probability hypothesis density filter," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pp. 491–498, IEEE, 2015.

[43] G. Costante, C. Forster, J. Delmerico, P. Valigi, and D. Scaramuzza, "Perception-aware path planning," *arXiv preprint arXiv:1605.04151*, 2016.

[44] P. Salaris, M. Cognetti, R. Spica, and P. R. Giordano, "Online optimal perception-aware trajectory generation," *IEEE Transactions on Robotics*, vol. 35, no. 6, pp. 1307–1322, 2019.

[45] J. Müller and G. S. Sukhatme, "Risk-aware trajectory generation with application to safe quadrotor landing," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3642–3648, IEEE, 2014.

[46] R. Senanayake, S. O'Callaghan, and F. Ramos, "Learning highly dynamic environments with stochastic variational inference," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2532–2539, IEEE, 2017.

[47] T. Krajník, J. P. Fentanes, J. M. Santos, and T. Duckett, "Fremen: Frequency map enhancement for long-term mobile robot autonomy in changing environments," *IEEE Transactions on Robotics*, vol. 33, no. 4, pp. 964–977, 2017.

[48] T. Krajník, T. Vintr, S. Molina, J. P. Fentanes, G. Cielniak, O. M. Mozos, G. Broughton, and T. Duckett, "Warped hypertime representations for long-term autonomy of mobile robots," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3310–3317, 2019.

[49] M. Phillips, B. J. Cohen, S. Chitta, and M. Likhachev, "E-graphs: Bootstrapping planning with experience graphs.," in *Robotics: Science and Systems*, vol. 5, p. 110, 2012.

[50] P. Lehner and A. Albu-Schäffer, "The repetition roadmap for repetitive constrained motion planning," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3884–3891, 2018.

[51] D. Coleman, I. A. Şucan, M. Moll, K. Okada, and N. Correll, "Experience-based planning with sparse roadmap spanners," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 900–905, IEEE, 2015.

[52] L. Nardi and C. Stachniss, "Experience-based path planning for mobile robots exploiting user preferences," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1170–1176, IEEE, 2016.

[53] L. Cohen, T. Uras, and S. Koenig, "Feasibility study: Using highways for bounded-suboptimal multi-agent path finding," in *Eighth Annual Symposium on Combinatorial Search*, 2015.

[54] M. Castillo-Lopez, P. Ludivig, S. A. Sajadi-Alamdari, J. L. Sanchez-Lopez, M. A. Olivares-Mendez, and H. Voos, "A real-time approach for chance-constrained motion planning with dynamic obstacles," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3620–3625, 2020.

[55] S. Aggarwal and N. Kumar, "Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges," *Computer Communications*, vol. 149, pp. 270–299, 2020.

[56] J. L. Sanchez-Lopez, M. Castillo-Lopez, M. A. Olivares-Mendez, and H. Voos, "Trajectory tracking for aerial robots: an optimization-based planning and control approach," *Journal of Intelligent & Robotic Systems*, vol. 100, no. 2, pp. 531–574, 2020.

[57] Y. Gheraibia, S. Kabir, K. Aslansefat, I. Sorokos, and Y. Papadopoulos, "Safety+ ai: a novel approach to update safety models using artificial intelligence," *IEEE Access*, vol. 7, pp. 135855–135869, 2019.

[58] S. Kabir, M. Walker, and Y. Papadopoulos, "Dynamic system safety analysis in hip-hops with petri nets and bayesian networks," *Safety science*, vol. 105, pp. 55–70, 2018.

[59] J. I. Aizpurua, V. M. Catterson, Y. Papadopoulos, F. Chiacchio, and G. Manno, "Improved dynamic dependability assessment through integration with prognostics," *IEEE Transactions on Reliability*, vol. 66, no. 3, pp. 893–913, 2017.

[60] A. Dheedan and Y. Papadopoulos, "Model-based distributed on-line safety monitoring," in *The Third International Conference on Emerging Network Intelligence (EMERGING 2011), Lisbon, Portugal*, pp. 1–7, 2011.

[61] D. Schneider and M. Trapp, "Conditional safety certification of open adaptive systems," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 8, no. 2, pp. 1–20, 2013.

[62] D. Schneider, R. Adler, P. Feth, J. Reich, and T. Braun, "Safety in cooperative automated systems,"

[63] S. Kabir, I. Sorokos, K. Aslansefat, Y. Papadopoulos, Y. Gheraibia, J. Reich, M. Saimler, and R. Wei, "A runtime safety analysis concept for open adaptive systems," in *International Symposium on Model-Based Safety and Assessment*, pp. 332–346, Springer, 2019.

[64] S. Patnaik, L. C. Jain, S. G. Tzafestas, G. Resconi, and A. Konar, *Innovations in Robot Mobility and Control*. Springer Berlin Heidelberg, 2005.

[65] G. Carbone, *Grasping in robotics*, vol. 10. Springer, 2012.

[66] D. Kragic and M. Vincze, *Vision for robotics*. Now Publishers Inc, 2009.

[67] B. D. Song, J. Kim, and J. R. Morrison, "Rolling horizon path planning of an autonomous system of uavs for persistent cooperative service: Milp formulation and efficient heuristics," *Journal of Intelligent & Robotic Systems*, vol. 84, no. 1-4, pp. 241–258, 2016.

[68] D3.1, "Deliverable d3.1 mrs design concept and methodology." Last accessed 16 December 2021.

[69] D1.1, "Deliverable d1.1 project requirements of sesame project." Last accessed 16 December 2021.