Project Number 101017258

D2.1 Specification of Multi-Robot Systems Capabilities

Version 1.1
15 March 2023
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

Project Partner Contact Information

Aero41 Frédéric Hemmeler Chemin de Mornex 3 1003 Lausanne Switzerland E-mail: frederic.hemmeler@aero41.ch	**ATB** Sebastian Scholze Wiener Strasse 1 28359 Bremen Germany E-mail: scholze@atb-bremen.de
AVL Martin Weinzerl Hans-List-Platz 1 8020 Graz Austria E-mail: martin.weinzerl@avl.com	**Bonn-Rhein-Sieg University** Nico Hochgeschwender Grantham-Allee 20 53757 Sankt Augustin Germany E-mail: nico.hochgeschwender@h-brs.de
Cyprus Civil Defence Eftychia Stokkou Cyprus Ministry of Interior 1453 Lefkosia Cyprus E-mail: estokkou@cd.moi.gov.cy	**Domaine Kox** Corinne Kox 6 Rue des Prés 5561 Remich Luxembourg E-mail: corinne@domainekox.lu
FORTH Sotiris Ioannidis N Plastira Str 100 70013 Heraklion Greece E-mail: sotiris@ics.forth.gr	**Fraunhofer IESE** Daniel Schneider Fraunhofer-Platz 1 67663 Kaiserslautern Germany E-mail: daniel.schneider@iese.fraunhofer.de
KIOS Panayiotis Kolios 1 Panepistimiou Avenue 2109 Aglatzia, Nicosia Cyprus E-mail: kolios.panayiotis@ucy.ac.cy	**KUKA Assembly & Test** Michael Laackmann Uhthoffstrasse 1 28757 Bremen Germany E-mail: michael.laackmann@kuka.com
Locomotec Sebastian Blumenthal Bergiusstrasse 15 86199 Augsburg Germany E-mail: blumenthal@locomotec.com	**Luxsense** Gilles Rock 85-87 Parc d'Activités 8303 Luxembourg Luxembourg E-mail: gilles.rock@luxsense.lu
The Open Group Scott Hansen Rond Point Schuman 6, 5th Floor 1040 Brussels Belgium E-mail: s.hansen@opengroup.org	**Technology Transfer Systems** Paolo Pedrazzoli Via Francesco d'Ovidio, 3 20131 Milano Italy E-mail: pedrazzoli@ttsnetwork.com
University of Hull Yiannis Papadopoulos Cottingham Road Hull HU6 7TQ United Kingdom E-mail: y.i.papadopoulos@hull.ac.uk	**University of Luxembourg** Miguel Olivares Mendez 2 Avenue de l'Universite 4365 Esch-sur-Alzette Luxembourg E-mail: miguel.olivaresmendez@uni.lu
University of York Simos Gerasimou & Nicholas Matragkas Deramore Lane York YO10 5GH United Kingdom E-mail: simos.gerasimou@york.ac.uk 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
1.0	QA review	23 December 2021
1.1	Comment Review	15 March 2023

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 assists the specification and configuration of robotic capabilities. 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 proposal [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

The deployment of Multi-Robot Systems (MRS) to support dependable missions in various real-world applications has earned significant attention due to advancements in robotics [2], [3], [4]. However, despite these advances, the configuration of such systems remains challenging due to uncertainties in the outcome of the operational models, rapid changes in environmental conditions, and emerging requirements. Moreover, MRS are often loosely connected enabling them to form and dissolve configurations dynamically. Also, available software and hardware components and heterogeneous robotic platforms, create increased variability levels. On the other hand, the lack of domain knowledge, unrealistic assumptions, and fragile design result in configurations that fail to accomplish the MRS mission [5], [6]. Unrealistic assumptions bias the expert's choice of a specific set of robot types, capabilities, and skills prior to thoroughly analyzing the operation goals proposed by the robotic user or use case. Even if the expert avoids these unrealistic assumptions and biases, the design team landed up to fragile design due to limited time or expertise dependency. Therefore, the design team must explore the large design space comprising numerous sufficiently robotic components that can be assembled in a 'plug-and-play' manner. The traditional methodologies, such as Domain Specific Language (DSL), are established based on the richness of expertise required to configure MRS missions [7]. For instance, in Semantic Robot Description Language (SRDL) [8], specific behavioral modeling and temporal logic languages are necessary, which imposes hierarchical dependencies between stakeholders and designers [9]. However, it is beneficial to provide generic and mission-oriented configurations, represented in a user-friendly robotics engineer methodology, that requires less domain expertise in modelling and reduces the dependency level [10]. Then, stakeholders can use configurations to tailor mission scenarios based on their domain expertise. 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 [11].

This document presents a stratagem, i.e., *"a generic, systematic capability-driven methodology for the dynamic configuration of MRS missions with variability level"*, to compose an 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 remphasise. 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. In summary the proposed novel stratagem

- enables robotic engineers to configure MRS missions.
- presents a set of robotic capability-related guiding steps for eliciting mission-specific interactions in terms of usecase scenarios.
- investigates viticulture usecase to evaluate its applicability. However, it can be applied to the other usecases.

1.1 Document Purpose

This document is 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.
- In Section 3, we introduce all the terminologies that we use throughout the report.
- Section4 describes the identification of MRS mission-level capabilities including a single robot-level and a team of robots.
- Section 5 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 skills and capabilities for a multi-robot systems mission-specific task.
- In Section 6 demonstrates a case study of autonomous pest management in viticulture. We demonstrate how the proposed stratagem can generate an MRS configuration for the mentioned use case scenario.
- Finally, Section 7 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.

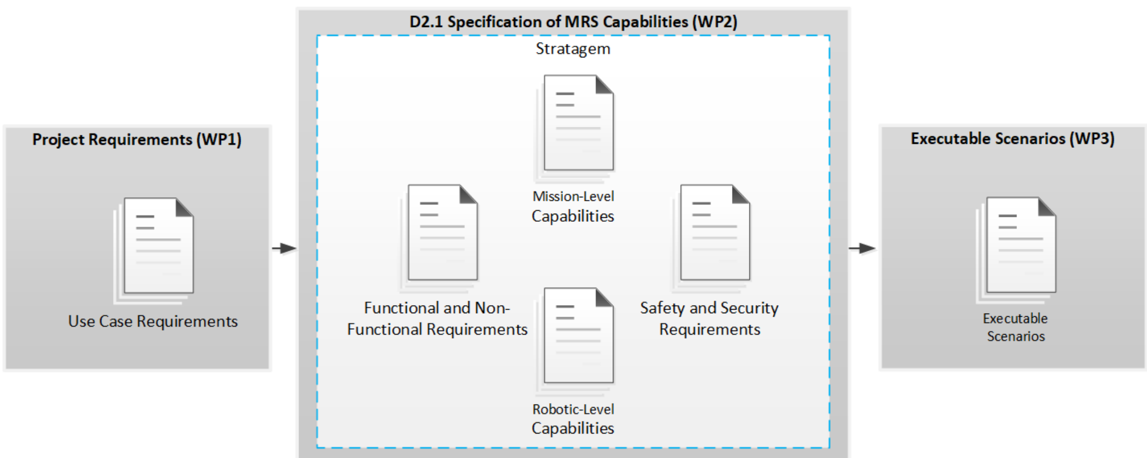

Figure 1: Workflow between the Project Requirements (WP1), Specification of MRS Capabilities (WP2), and Specification of Executable Scenarios (WP3). The dotted blue rectangle illustrates the focuses of this deliverable.

2 Background and Motivation

This section discusses the related work in the specification MRS. Primarily, the existing works, related to MRS specifications can be categorized as follows.

Several methods offer user-oriented approaches to the configuration of MRS missions based on programming languages, which are specific to application domains, target platforms, and end-users [12], [13]. Similarly, Graphical User Interface (GUI)-oriented, Robotic Modeling Language (RobotML) [14], and FLYAQ [15] approaches are tightly bound to the application, even though they easily configure the MRS missions in a graphical manner. In contrast, task-oriented ones are based on markup languages [16]. Nevertheless, the programming-based languages provide a richer feature set and a higher degree of expressiveness. However, these approaches require programming skills. On the other hand, markup-based languages come at the cost of reduced flexibility concerning the application domains and platforms.

Flowchart-oriented approaches configure the mission based on an activity diagram, which is easy to configure [17]. However, it lacks concurrency support. As a solution, statechart-oriented approaches consider the mission based on state diagrams, which is suitable for robotic missions [18]. This requires a detailed definition of MRS configurations to achieve a mission, which may become complicated in practical cases. Another approach is Petri Net Plans (PNPs), which enable developers to describe plans for MRS missions [19]. Despite their expressive power, PNPs require a precise definition of every action of a robot in an MRS mission.

Temporal logic-oriented approaches, particularly Linear Temporal Logic (LTL) and Computation Tree Logic (CTL), are based on temporal logic models, which allow users to analyze mission accomplishment, using the model checker [20]. Even with automatic generation of mission specifications e.g., [21], these approaches are complex in terms of manual writing and prone to error.

Finally, there exist some new approaches, using skill-based or component-based modeling, to overcome the drawbacks of the above methods, e.g., [2], [3], [4], [22]. However, this requires programming language knowledge and high-level abstraction to the robotic users, which might impose the reality gap between the conceptual design and deployment.

Motivated by the aforementioned considerations, our proposed *stratagem* methodology supports the ExSce specifications using automated model transformation, where programming skills are not required. This is done by describing generic and reusable *capabilities* that can configure complex missions with increased domain flexibility. For this, we propose a set of *capabilities* that allow configuring of concurrent and collaborative missions. In our *stratagem*, we use the domain expert's knowledge to rank the robotic-level *capabilities* and to help robotics engineers in decision-making, configuring practical MRS missions. The knowledge of inter-relationships between various decisions is an essential part of domain expert's knowledge. Therefore, this knowledge is considered a capability-driven methodology for the configuration of collaborative MRS missions.

3 Definition of Terms

We use standard robotic terminologies [23], [24], [25], [26] with an example as given below:

Example: Picking up an object is an ordered sequence of actions, such as i) recognizing the object, ii) moving the arm towards it, iii) grasping it, and finally, iv) lifting it.

- *Robot*: A robot is a physical agent that performs actions by manipulating the physical world.
- *Action*: An action is a process initiated by an agent and capable of changing the world. For example, *PickingUpAnObject* is an action in a robotic arm mission.
- *Capability*: A capability is a property of a robot that allows it to perform or execute a certain type of action. For example, action *PickingUpAnObject* needs *ObjectRecognitionCapability*, *MovingArm-Capability*, and *GraspingCapability* to perform the picking action. A robot has a capability due to its components.

- *Component*: A component of a robot is hardware, software, and information objects. The hardware components are sensors and actuators, the software components are programmatically implemented algorithms, and the information objects are knowledge bases. For example, capability *ObjectRecognitionCapability* needs *VisualSensor*, and *ObjectRecognitionAlgorithm*.
- *Behavior*: A robot's behaviour is a property that makes it perform a certain action in a way (such as autonomous, manual, or semi-autonomous) when it encounters a certain situation. Robotic Behavior is the composition of capabilities and skills to realise a robot's action. The relation between capability and behavior is how a robot utilize its capability when it encounters a certain class of situations. For example, autonomous obstacle avoidance for a robot encounters a class of situation 'Facing obstacle', which requires 'Avoiding Obstacle' capability with 'Autonomous' behavior of the robot.
- *Skill*: A skill provides access to functionalities realized by robotic components. For example, in a viticulture scenario, a spraying drone has the skill of 'spraying' pesticides. However, several robots have different types of skills. After analysing the use cases, the class of skill sets are defined as follows, but not limited to the considered use cases.
  - Data collection: It refers to collecting raw data from an environment. For example, a drone can collect thermal images from an area.
  - Inspection: The inspection class refers to inspecting objects or areas.
  - Manipulation: This class of skill refers to interacting physically with objects, e.g., stacking, assembling, handling, etc.
  - Dispensing: It refers to applying external material to objects or environment, e.g., glueing, welding, painting, spraying, disinfecting, etc.
- *Robotics engineer*: A robotics engineer is a person who is responsible for configuring the missions, considering hardware, software, and required capabilities.
- *Safety engineer*: A safety engineer is a person who is responsible for defining safety-related aspects. A safety engineer explores ways for generating safety parameters to Computer-Aided Design (CAD) and kinematic models and feasibility studies. This is expected to provide a basis for Executable Digital Dependability Identities (EDDI) model-based artifacts that carry verifiable dependability models of their reference robotic systems produced at design time [27]. In addition, it captures safety and security hazards, their causes, effects, and possible corrective actions. The objective is to simplify hazard identification and causal analysis and improve the efficiency of safety assessments.
- *Security engineer*: A security engineer is a person who is responsible for defining security-related aspects. Using EDDI to instrument the robotic team with security models enables the team to anticipate potential malfunctions or external cyber-attacks and initiate mitigation actions to bring the affected robot to a safe stop [27].
- *Domain experts*: People with extensive robotic knowledge of individual-level and team-level missions, such as robotics engineers, safety engineers, and security engineers.
- *Knowledge base*: A knowledge base entity contains reusable information about robotic capabilities.
- *Use case*: A use case is an industrial entity that describes the interaction with a system yet to be designed, concentrating on what people try to achieve rather than how they achieve it, such as i) Dependable multi-robot systems in battery innovation centre, ii) Disinfecting hospital environments using robotic teams, iii) Power station inspection using autonomous multi-robot systems, iv) Autonomous pest management in viticulture, and v) Security management of MRS-based assembly lines.

Based on these definitions, the following points are worth mentioning. A robot has a capability with associated components. It is difficult to compare a robot and an action directly because they have different characteristics, i.e., a robot has physical and spatial characteristics, whereas an action is abstract and hierarchical. Therefore, the capability is introduced to the link between robots and their actions. Action depends on a set of capabilities and equipped components of a robot. In order to exhibit a particular capability, a robot has to possess a set of components that cooperate and jointly enable the robot to exhibit the capability. Use case applications in different domains might require similar capability of robots. From the perspective of end-users, this textit robotic capabilities are modelled as *robotic features*, which include the nominal functionality, encapsulated as *operational scenarios* [28].

4 Identification of MRS Capabilities

This section describes the identified MRS capabilities in terms of mission-level, including robotic individual-level and robotic team-level missions. In order to exhibit a particular capability, a robot has to possess a set of components. Accordingly, capability-associated components are explained, as illustrated in Figure 2. We purposefully divided the capabilities into an individual level and an MRS level. Individual-level capabilities refer to the ability or capability a single robot has. For example, as described in WP 2.1, grasping, perception, navigation, mobility (2D or 3D), etc. However, the limited resources of a single robot can not perform all the tasks for a large complex mission. Additionally, there could be certain scenarios when a robot can not use some sensors due to malfunction. But, this problem can be solved by collaborative sensor fusion using multiple robots. Therefore, multiple heterogeneous robots come into the picture to execute a big complex mission collaboratively. Hence, we define team-level capability, as described in the WP 2.1, such as collaborative perception, collaborative navigation and sensor fusion, and collaborative intelligence. In reality, individual capability or team-level capability can be composed in several ways based on the robot types and behaviour. This report describes all the capabilities based on the proposed five use case scenarios.

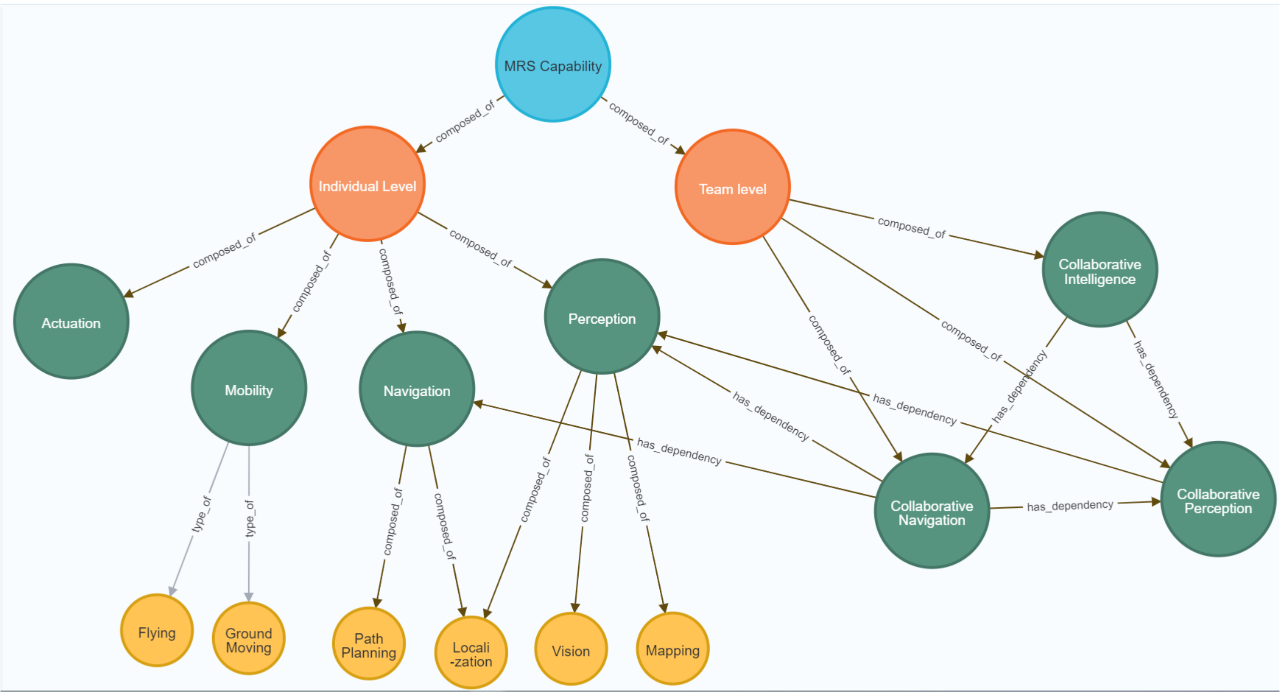

Figure 2: Capability-Associated Components.

4.1 Individual-Level Capabilities

4.1.1 Mobility

Robots can have the mobility to move physically in the environment. Two mobility classes are identified considering the five use cases, i) flying, e.g., Unmanned Aerial Vehicles (UAVs), and ii) ground moving, e.g., Autonomous Ground Vehicles (AGVs) and Robotic Arms (ARMs).

4.1.2 Perception

Perception is the primary capability enabling robots to reason and make decisions while interacting with the environment. This capability refers to sensory perception of the environment to extract information. The perception enables robots to interact with the environment in various ways, e.g., obstacle avoidance. The perception highly depends on the sensors' availability to acquire the data from the environment. These sensory data enable a robot for vision, mapping, and localization.

4.1.3 Navigation

Navigation enables the robot's mission-level path or trajectory towards the goal safely, given the perceptive information. The crucial components of a robot's navigation are localization, i.e., position and orientation and safe path planning i.e., motion between the current position to a goal position with respect to a reference frame. To realize this, the *perception* and *mobility* capabilities, as well as the trajectory planning and trajectory tracking components are required.

4.2 Team-Level Capabilities

4.2.1 Collaborative perception

This uses multiple robots acting as viewpoints to improve and gather knowledge about the environment and themselves. The knowledge could be obstacles, other robots, potential hazards or threats, and uncertainties. This is because collaborative MRS application requires a rich, consistent, and accurate understanding of the environment and robot state. On the other hand, a single robot frequently suffers from sensor limitations (e.g., range, occlusion), while a robotic team can combine multiple observations and share their results. Therefore, the capability to collect and fuse the sensor information from MRS improves the overall perceptive and, situational awareness information. For this capability, all the robots in the team rely on the individual-level *perception* capability. Also, Robot-to-Robot (R2R) communication component is required to communicate between robots resulting better understanding of the environment and safe navigation.

4.2.2 Collaborative navigation

The collaborative navigation capability comprises the planning, tracking, and sharing of situational and perceptional information among the MRS members to reach individual goals safely, given mission tasks. Planning is to find an optimal path or trajectory to the goal, taking into account gathered collaborative perceptive and situational information for a team of robots. Consequently, required actuator actions are designed and commanded to each robot to track the path as closely as possible. The collaborative aspect is incorporated in both planning and tracking components, by relying on the information provided not only from each robot but also from the other MRS members. To realize this capability, all the robots in the team rely on the *collaborative perception* capability and individual-level *navigation* capability.

4.2.3 Collaborative intelligence

Collaborative intelligence enables the decomposition of complex tasks into smaller tasks manageable by individual robots, creating coalitions within an MRS and distributing tasks between team members. To achieve this capability, all the robots in the team require the *corealisetive perception* capability, *collaborative navigation* capability, and *knowledge base component.*

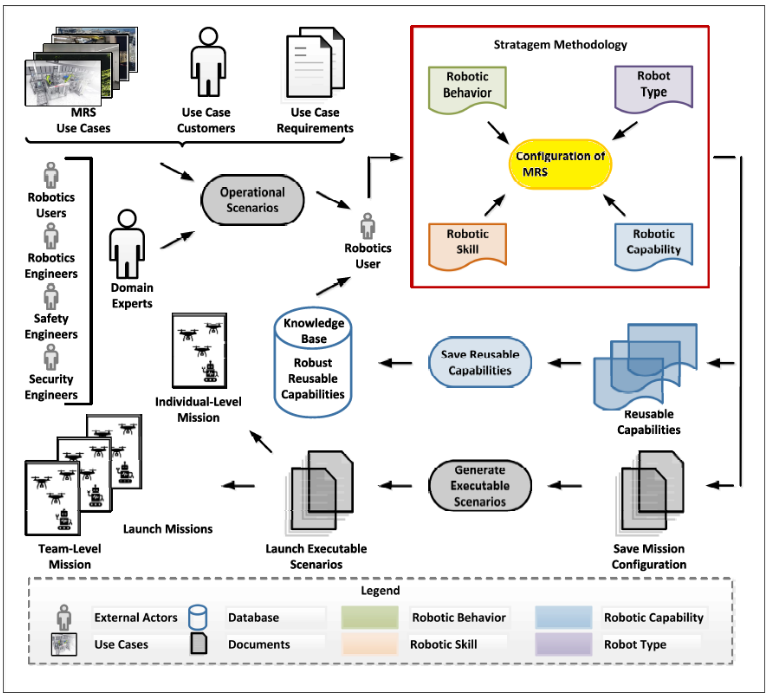

Figure 3: Stratagem: A Capability-Driven Configuration of Collaborative Missions for Multi-Robot Systems.

5 Proposed Stratagem Methodology

This section elaborates on the proposed stratagem methodology considering Figure 3, the definitions and identified capabilities, given in Sections 3 and 4, respectively. The proposed stratagem aims at addressing two main challenges, identified in the literature in Section 2, namely, removing unrealistic assumptions and fragile MRS designs. As per the proposal, unrealistic assumptions impose biases by the expert towards the selection/configuration of a specific set of robot types, capabilities and skills, before analysing thoroughly the operation goals, proposed by the robotic user or use case. For instance, for handling large and heavy boxes in an assembly line, one trivial solution is the use of robotic manipulators. The expert might have suggested this to the user, which totally biases the user towards implementing the manipulator. However, given the user's needs and goals, via a thorough analysis of the operational environment, other types might come into the picture, e.g., a ground robot equipped with a hydraulic jack. Even if the expert avoids these unrealistic assumptions and biases, this way of specifying the MRS operation imposes another challenge, which is, namely, a fragile MRS design, due to the limited time or expertise dependency, the design team has to sufficiently explore the large design space comprising numerous robotic components that can be put together in a 'plug-and-play' manner. Therefore, our proposed stratagem addresses these challenges, i.e., we foresee and analyse the whole MRS specification process to come up with a methodology with a reduced requirement of expert knowledge, as well as to avoid the mentioned fragile design via a systematic way of the specification process.

It is worth noting that the output of this stratagem is a set of MRS configurations, addressing the aims and needs of the operations, including the identified robot types, MRS requirements, capabilities, skills and behaviours. This specified configuration set is sent to the ExSce to determine the operation scenario.

The proposed stratagem methodology supports both robotic individual-level and team-level configurations. After the configuration is fulfilled, the robotics engineer generates the executable scenarios based on the saved mission configuration, and launches them on the corresponding framework, i.e., simulation or real world. Initially, the Skill-based Capability Rates Matrix (SCRM) is to be constructed using the knowledge from the various domain experts, including robotic engineers, safety, and security engineers. SCRM comprises the rating for different robot types, considering the identified skills and corresponding capabilities, both at individual and team levels. This helps select the proper robot type and configuration for the respective robotic mission. In SCRM, the capabilities are considered instead of components. Hence, this methodology enables the robotics engineer to configure the missions dynamically and reliably with improved variability levels, i.e., the proposed stratagem can be applied for different use case scenarios without the need for any modifications or changes in the core of methodology.

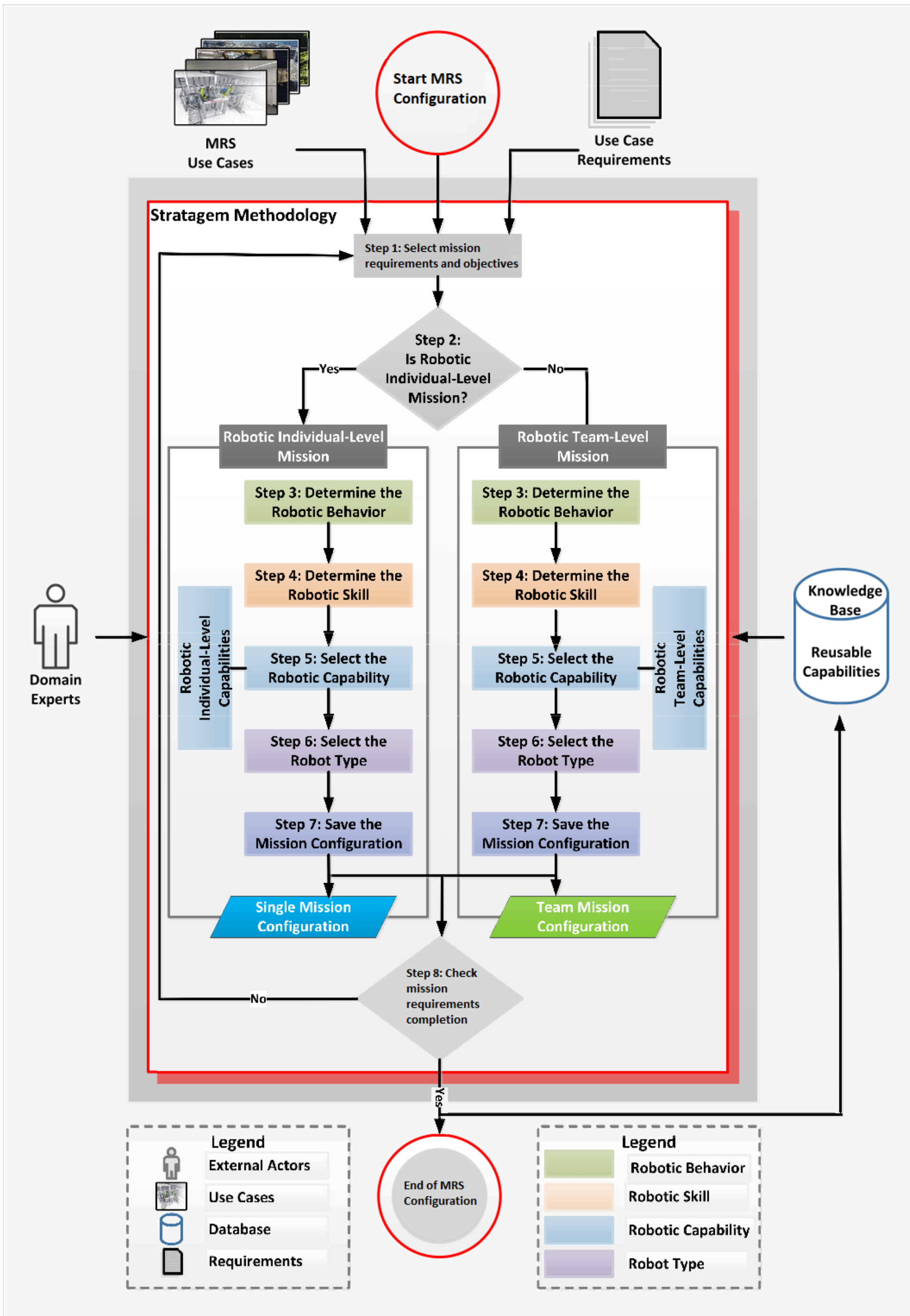

Figure 4: Stratagem Methodology to Configure MRS Missions.

It is worth noting that the set of mission objectives is defined based on the use case requirements. This is done by use case customers with the help of domain experts. To this end, the sequence of steps of the proposed

Algorithm 1 MRS Mission Configuration

Inputs: Mission Objectives
$Configuration \leftarrow \emptyset$
for $Objective \in \{\text{Objective}_1, \ldots, Objective_n\}$ **do**
  if $Objective$ is met at individual-level (Step 2) **then**
    $B \leftarrow Behaviour \in \{Autonomous, Semi-Autonomous, Manual\}$(Step 3)
    $S \leftarrow Skill \in \{DataCollection, Inspection, Manipulation, Dispensing\}$ (Step 4)
    $C_I \leftarrow Capability \in \{Mobility, Perception, Navigation\}$(Step 5)
    $T \leftarrow \arg \max_{RobotType \in \{AGV, UAV, ARM\}} SCRM(S, C_I)$
    $Configuration \leftarrow Append(B, S, C_I, T)$
  else ▷ $Objective$ is met at team-level (Step 2)
    $B \leftarrow Behaviour \in \{Autonomous, Semi-Autonomous, Manual\}$(Step 3)
    $S \leftarrow Skill \in \{DataCollection, Inspection, Manipulation, Dispensing\}$ (Step 4)
    $C_T \leftarrow Capability \in \{Collab.Perception, Collab.Navigation, Collab.Intelligence\}$
    $C_I \leftarrow \left\{Capability \in \{Mobility, Perception, Navigation\} | Capability \subseteq C_T\right\}$
    $T \leftarrow \arg \max_{RobotType \in \{AGV, UAV, ARM\}} SCRM(S, C_I)$
    $Configuration \leftarrow Append(B, S, C_I, C_T, T)$
  end if
end for
Output: $Configuration$

stratagem methodology is illustrated in Figure 4 and explained in detail as follows, and accordingly Algorithm 1 is given.

Step 1: Select mission requirements and objectives

The robotics engineer selects one of the provided missions for a given use case. The mission requirements, objectives, pre-conditions, main scenarios, and constraints are derived based on the understanding of user needs as well as the business environment that drives many of the development and architectural decisions for using MRS for applications in each domain. This is an input to the proposed stratagem, as described in Algorithm 1 and Figure 4.

The robotic engineers require some general guidelines to avoid biases in the specification process. Therefore, we propose a generic SCRM, given in Table 2, to address the aforementioned issue. It is worth noting, this SCRM is based on the extensive analysis of the use cases, but it is not restricted only to these five use cases, given in Table 1.

Table 1: Selected use cases of SESAME.

Use Cases	Stakeholders	Domain	Missions
Use Case 1: Autonomous Pest Management	Domaine Kox, Aero41, and LuxSense	Viticulture	Multi-UAVs
Use Case 2: Disinfecting Hospital Environments	Locomotec	Healthcare	Multi-AGVs
Use Case 3: Autonomous Power Station Inspection	Cyprus Civil Defence	Inspection	Multi-UAVs
Use Case 4: Battery Innovation Centre	AVL List GmbH	Battery	AGVs and ARMs
Use Case 5: Security Management in Assembly Lines	KUKA Assembly and Test	Robotic	Multi-ARMs

Step 2: Initial identification of mission-level

Table 2: Evaluation of Skills based on Robotic Individual-Level Capabilities.

Skill-Based Capability Rate Matrix (SCRM)				
Robotic Skils	**Individual-Level Capability**	**Rate of Robot Type**		
		AGV	**UAV**	**ARM**
Data Collection	Mobility	Medium	High	Low
	Perception	Medium	High	Low
	Navigation	Medium	High	Low
Inspection	Mobility	Medium	High	Low
	Perception	Medium	High	Low
	Navigation	Medium	High	Low
Manipulation	Mobility	Medium	Low	High
	Perception	Medium	Low	High
	Navigation	Medium	Low	High
Dispensing	Mobility	Medium	High	Low
	Perception	Low	Medium	High
	Navigation	Low	Medium	High

The robotics engineer initially and intuitively identifies if the mission is at robotic *individual-level* or *team-level* mission. However, this is not the final consideration, as we might modify this later on throughout the proposed stratagem, considering the other given/identified objectives.

If the mission requirement is directly provided as an MRS, then the proposed stratagem proceeds considering team level in the next steps. However, as a rule of thumb, we propose to start with individual-level configuration and then later on the proposed stratagem checks if MRS is required or not, after the first iteration.

Step 3: Determine the robotic behavior

The robotic behavior is determined based on the mission requirements. The robotic behavior includes *autonomous*, *semi-autonomous*, and *manual*. For example, if the requirement describes the MRS is fully autonomous or human intervention, that is semi-autonomous, or fully manual, then the stratagem determines the behaviour of the system.

Step 4: Determine the robotic skill

The robotics engineer determines the required robotic skills, out of the above-mentioned classes, based on the selected mission description and robotic behaviour.

After analysing the identified skills, the dependent capabilities, are identified as follows:

- Data collection: This skill is considered as collecting data from the environment or other robots and humans. For this skill, at the individual level, the capabilities of perception and navigation are mandatory. Also, mobility is optional which can improve the data collection skill. On the other hand, at the team-level, collaborative perception capability is required.
- Inspection: This skill is considered as collecting, detecting or monitoring an object or area or human. For this skill, at the individual level, the capabilities of perception and navigation are mandatory. Also, mobility is optional which can improve the data collection skill. On the other hand, at the team level, collaborative perception and collaborative intelligence capabilities are required.
- Manipulation: This skill is considered as affecting the environment physically and directly by the robot. For this skill, at the individual level, the capabilities of mobility, perception and navigation are mandatory. At the team-level, collaborative perception and collaborative capabilities are mandatory and collaborative intelligence is optional.
- Dispensing: This skill is considered as applying external material on a specific object or area. At the individual level, the capabilities of mobility, perception and navigation are mandatory. At the team-level,

collaborative perception and collaborative capabilities are mandatory and collaborative intelligence is optional.

Step 5: Select the robotic capability

The robotics engineer selects the robotic capabilities based on the determined skills. This is done considering *Step 2*, i.e., if robotic individual-level or team-level capabilities are required. If the mission is at the individual-level, then the capabilities, given in Section 4.1, are determined based on SCRM in Table 2.

If the mission is at team-level, then the corresponding team-level capabilities of Section 4.2 are considered, which further determine the corresponding individual-level capabilities. Then, we follow SCRM Table 2 to determine the robot types. Also, the selected capability implies the associated components, as explained in Section 4, e.g., knowledge base component.

Step 6: Select the robot type

The robotics engineer selects the type based on the selected capabilities, considering the highest rate amongst the type for the associated capability. Considering various use cases, we have identified the robot types as *AGV*, *UAV*, and *ARM*, as mentioned in Table 2.

Step 7: Save the mission configuration

The robotics engineer saves the configuration of the mission at both individual and team-level, in terms of Behaviour, Skill, capability (both individual and team-levels), and type of robots, as described in algorithm 1.

Step 8: Check mission requirements completion

The robotics engineer checks if all the mission requirements/objectives are fulfilled. If not, *Step 2* to *Step 7* are repeated for the other ones. Otherwise, the configuration of the mission is fulfilled. The extracted capabilities from the mission configuration is stored in the knowledge base for similar configurations.

The output is the saved configurations, which are sent to the ExSce.

Figure 5: Autonomous Pest Management System in Viticulture Use Case.

6 Case Study: A Collaborative MRS Mission for Autonomous Pest Management System in Viticulture

This section applies the proposed stratagem methodology for configuring autonomous pest management in the viticulture use case mission. Here, we describe the MRS mission descriptions and the requirements first. Then we proceed further steps according to the proposed stratagem, as described in Algorithms 1 and Figure 4.

6.1 MRS Mission Description

In this use case, the MRS system completes a mission through the collaboration of two UAVs with two different tasks (inspection and spraying), as shown in Figure 5. The MRS has the sufficient processing power to execute collaborative MRS capabilities onboard and is equipped with multiple sensors. The first UAV is used for detecting and helping the spraying UAV to the generation of an autonomous fungicide spraying mission in the vineyard. The second UAV is used for the autonomous fungicide spraying mission in the vineyard. For both tasks (inspection and spraying), UAVs are deployed for real-time inspection, fungicide spraying, and coordination purposes. This inspection UAV should be able to take flight at least the same time as the spraying UAV to monitor the whole mission without being a bottleneck. In addition, it carries Camera and Inertial Measurement Unit (IMU) sensors to perceive the environment and evaluate the state of the mission. During

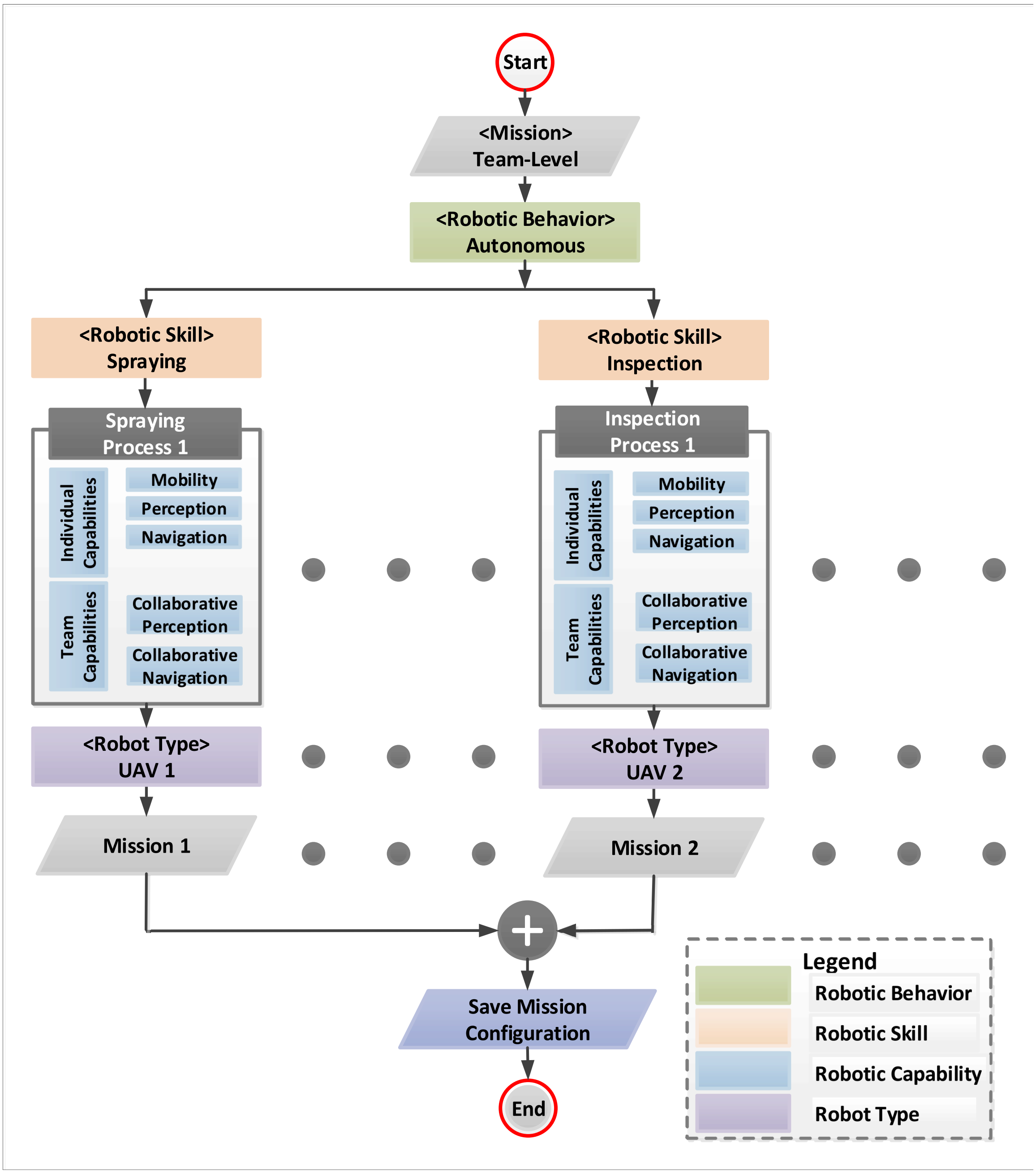

Figure 6: A Basic System Flow for Configuring the MRS Mission within the Viticulture Use Case for Autonomous Pest Management.

fungicide spraying, the spraying UAV needs to navigate toward an infection hotspot and apply a variable, pre-defined rate of fungicides. This rate can vary depending on the grade of infection. Therefore, high importance is placed on the adequate amount of fungicide applied to a specific hotspot. To this end, the spraying UAV needs precise positioning accuracy to maintain a low but constant distance from the vines. As the efficiency and security of the fungicide spraying rely on high positioning accuracy, a fall-back system of collaboration perception, sensor fusion, trajectory planning, and trajectory tracking is deployed on the inspection UAV in case of loss of positioning accuracy.

The summary of mission requirements is given as follows.

Use Case : A Collaborative MRS Mission for Autonomous Pest Management System in Viticulture.

Description: Multi-Robot Systems (MRS) is deployed for autonomous spraying fungicide and inspection of spraying fungicide operation in the vineyard.

Pre-Conditions: The first set of Unmanned Aerial Vehicles (UAVs) is equipped with an adequate amount of fungicide spray and requires onboard sensors to navigate toward an infection hotspot to spray. The second set of UAVs is equipped with visual and IMU sensors to inspect the spraying fungicide mission state.

Main Scenario:

1. The first set of UAVs take off and commence the fungicide spraying.
2. The second set of UAVs take off and continuously inspect the spraying mission state in the vineyard.
3. These two sets of UAVs should fly simultaneously.
4. Both sets track and send the location and state of each UAV in the corresponding set to the Ground Control Station (GCS).

Constraints: In the case of any UAV that has not had a sufficient battery or adequate amount of fungicide to continue the mission, a safe Return To Home (RTH) is enabled to return to the base station quickly.

6.2 Configuration of Autonomous Pest Management System Using Proposed Stratagem

A basic system flow for configuring the MRS mission within the viticulture use case for autonomous pest management has been shown in Figure 6 applying the proposed stratagem, as described in Algorithms 1 and Figure 4, and described below:

Given the above missions, this mission is at the team-level and required to be autonomous. For this mission, two classes of skills are identified as described in Section 3 and Table 2, which are dispensing in the form of spraying and inspection in the form of detecting another drone. Based on the identified classes of skills, the associated required capabilities are mobility, perception, navigation, collaborative perception, and collaborative navigation, as described in Section 5. Now, considering required capabilities at the individual-level, with Table 2, the robot type is selected as UAV, as it has the highest rate for the corresponding individual-level capabilities. Then, this configuration is saved as stated below:

$$
\begin{aligned}
Configuration = \{ & Behaviour : Autonomous, \\
& Skill : [Dispensing : Spraying, Inspection], \\
& IndividualCapability : [Mobility : Flying, Perception, Navigation], \\
& TeamCapability : [CollaborativePerception, \\
& \qquad CollaborativeNavigation, \\
& \qquad CollaborativeIntelligence], \\
& Type : UAV \}
\end{aligned}
$$

collaborative mission, and the executable scenario is generated and launched. It should be noted that based on the selected capabilities, the associated components are determined, including vision, mapping, localization, trajectory planning, trajectory tracking, and R2R communication.

7 Conclusions

This deliverable proposed a novel stratagem methodology for Multi-Robot System mission configuration, including robotic individual-level and team-level capabilities, which were extracted by the domain experts. The proposed stratagem encapsulated high-level robotic capabilities, skills, behaviours, and types. We presented an algorithm to elaborate the proposed stratagem logically. We have demonstrated its use through the systematical configuration of MRS for an industrial use case. In the proposed stratagem, it is aimed at enabling engineers to specify and configure the components and the required capabilities, as well as describing high-level robotic skills and behaviours based on the type of the robot. In addition to individual robot capabilities, the proposed stratagem tackles the configuration at the team-level, with collaorative perception and navigation. Finally, the stratagem conforms to public metamodels and it is reusable for similar applications.

References

[1] TOG, "Sesame: Secure and safe multi-robot systems project proposal," 2020. Last accessed 16 December 2021.

[2] Y. Pane, E. Aertbeliën, J. D. Schutter, and W. Decré, "Skill-based programming framework for composable reactive robot behaviors," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7087–7094, 2020.

[3] C. Lesire, D. Doose, and C. Grand, "Formalization of robot skills with descriptive and operational models," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7227–7232, 2020.

[4] F. Nägele, L. Halt, P. Tenbrock, and A. Pott, "A prototype-based skill model for specifying robotic assembly tasks," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 558–565, 2018.

[5] L. E. Parker, D. Rus, and G. S. Sukhatme, "Multiple mobile robot systems," in *Springer Handbook of Robotics*, pp. 1335–1384, Springer, 2016.

[6] S. Robotics, "Robotics 2020 multi-annual roadmap for robotics in europe," in *SPARC Robotics*, p. 2018, EU-Robotics AISBL, The Hauge, The Netherlands, 2016.

[7] A. Nordmann, N. Hochgeschwender, D. Wigand, and S. Wrede, "A survey on domain-specific modeling and languages in robotics," 2016.

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

[9] C. Heinzemann and R. Lange, "vtsl - a formally verifiable dsl for specifying robot tasks," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8308–8314, 2018.

[10] 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.

[11] SESAME, "Sesame: Secure and safe multi-robot systems," 2020. Last accessed 16 December 2021.

[12] D. Vanthienen, M. Klotzbu, J. De Schutter, T. De Laet, H. Bruyninckx, *et al.*, "Rapid application development of constrained-based task modelling and execution using domain specific languages," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1860–1866, IEEE, 2013.

[13] 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.

[14] 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.

[15] 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.

[16] E. Aertbeliën and J. De Schutter, "etasl/etc: A constraint-based task specification language and robot controller using expression graphs," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1540–1546, 2014.

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

[18] 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.

[19] 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.

[20] 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.

[21] 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.

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

[23] J. I. Olszewska, M. Barreto, J. Bermejo-Alonso, J. Carbonera, A. Chibani, S. Fiorini, P. Goncalves, M. Habib, A. Khamis, A. Olivares, E. P. de Freitas, E. Prestes, S. V. Ragavan, S. Redfield, R. Sanz, B. Spencer, and H. Li, "Ontology for autonomous robotics," in *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 189–194, 2017.

[24] E. Pignaton de Freitas, J. I. Olszewska, J. L. Carbonera, S. R. Fiorini, A. Khamis, S. V. Ragavan, M. E. Barreto, E. Prestes, M. K. Habib, S. Redfield, A. Chibani, P. Goncalves, J. Bermejo-Alonso, R. Sanz, E. Tosello, A. Olivares-Alarcos, A. A. Konzen, J. Quintas, and H. Li, "Ontological concepts for information sharing in cloud robotics," *Journal of Ambient Intelligence and Humanized Computing*, vol. 43, 06 2020.

[25] J. Bermejo-Alonso, C. Hernández, and R. Sanz, "Model-based engineering of autonomous systems using ontologies and metamodels," in *2016 IEEE International Symposium on Systems Engineering (ISSE)*, pp. 1–8, 2016.

[26] M. Askarpour, C. Tsigkanos, C. Menghi, R. Calinescu, P. Pelliccione, S. García, R. Caldas, T. J. von Oertzen, M. Wimmer, L. Berardinelli, M. Rossi, M. M. Bersani, and G. S. Rodrigues, "Robomax: Robotic mission adaptation exemplars," in *2021 International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, pp. 245–251, 2021.

[27] K. Aslansefat, P. Nikolaou, M. Walker, M. N. Akram, I. Sorokos, J. Reich, P. Kolios, M. K. Michael, T. Theocharides, G. Ellinas, D. Schneider, and Y. Papadopoulos, "Safedrones: Real-time reliability evaluation of uavs using executable digital dependable identities," in *Model-Based Safety and Assessment* (C. Seguin, M. Zeller, and T. Prosvirnova, eds.), (Cham), pp. 252–266, Springer International Publishing, 2022.

[28] 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.