D2.2 - FINAL VISION SCENARIOS AND USE CASE DEFINITION

Deliverable ID	**D2.2**
Deliverable Title	**Final Vision Scenarios and Use Case Definition**
Work Package	**WP2**

Dissemination Level	**PUBLIC**

Version	**1.0**
Date	**2018-07-04**
Status	**Final**

Lead Editor	**Rafael Lopez (ROBOTNIK)**
Main Contributors	Sarah Suleri, René Reiners **(FRAUNHOFER),** Artiza Elosegui, Edin Arnautovic **(TTTECH),** Omar Morando **(DGSKY),** Alessandra Bagnato, Etienne Brosse **(SOFTEAM),** Melanie Schranz **(LAKE),** Ángel Soriano **(ROBOTNIK)**

Published by the CSPwarm Consortium

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731946.

Document History

Version	Date	Author(s)	Description
0.1	2018-04-20	Rafael López (ROBOTNIK)	First Draft with relevant materials
0.2	2018-04-27	Rafael López (ROBOTNIK)	Comments added
0.3	2018-05-4	Angel Soriano (ROBOTNIK)	New material added
0.4	2018-05-18	Rafael López (ROBOTNIK)	Comments included
0.5	2018-06-28	Ángel Soriano (ROBOTNIK)	Rewrite the logistic scenario
0.6	2018-07-04	René Reiners (FIT)	Added CPSwarm Workbench usage scenario Merged scenario inputs Finalized overall scenarios.
1.0	2018-07-06	Rafael López (ROBOTNIK)	Final version ready to be submitted to EC

Internal Review History

Review Date	Reviewer	Summary of Comments
09-05-2018	Wilfried Elmenreich (UNIKLU)	Comments and few modifications inserted.
28-05-2018	Alessandra Bagnato (SOFTEAM)	Comments and few modifications inserted.

Executive Summary

The present document is a deliverable of the CPSwarm project, funded by the European Commission's Directorate-General for Research and Innovation (DG RTD), under its Horizon 2020 Research and innovation program (H2020), reporting the results of the activities carried out by WP2 - Use cases, requirements engineering and business models. The main objective of the CPSwarm project is to develop a workbench that aims to fully design, develop, validate and deploy engineered swarm solutions. More specifically, the project revolves around three vision scenarios; Swarm Drones, Swarm Logistics Assistant and Automotive CPS. The scenarios were outlined in the proposal and are refined within the engineering efforts alongside the project, driven by WP2.

D2.1 reported the iterative process of revision, extension, refinement and maturation process of the set of use cases grouped with respect to the involved actors. The use cases specified in this deliverable should be seen as a final vision towards which the project will reach.

Deliverable D2.2 is an update of the deliverable D2.1 finalizing in particular the vision scenarios as thematic foundation for the activities of the project. In addition, an overarching scenario describe the general usage of the CPSwarm Workbench in order to realize the vision scenarios.

The visions are used for inter-project communication in order to focus on the development scenarios and help to centralize the process of thinking about the implementation of those use cases. Besides the aspects regarding development, the vision scenarios are used as understandable story for externally communicating the project's aims and tell the audience what kinds of application can be designed with the CPSwarm Workbench, the actual outcome of the project.
Furthermore, this deliverable formulates the foundation for the final requirement analysis to be documented in D2.7 in WP2, and later for the remaining technical WPs (WP3 to WP7), towards the demonstration (WP8).

Table of Contents

Contents

1 Introduction

This deliverable documents the results of Task 2.1 *Vision scenarios, use cases and initial requirements*. The purpose of this deliverable is to update the vision scenarios that were described at the deliverable D.2.1 as an initial vision of them. The use case specification will be provided for:

- Swarm Drones
- Autonomous Freight Vehicles
- Swarm Logistics Assistant

This document describes the activities to support the identified workbench workflow, adapting it to the different environments involved in the CPSwarm project and provides a thorough analysis of the final use cases. These high-level procedures and the use cases will guide the development phases within the technical work packages, and therefore, this deliverable will be a common reference point for the CPSwarm consortium with relevance to architectural (WP3) questions and impacts on implementation (WP7 and WP8) as well as exploitation (WP9) efforts.

The **main objectives** of the activities that were performed by Task T2.1 so far are listed in the following:

- Definition of the methodology for analysing the stakeholders
- Determination of stakeholder categories
- High-level domain analysis of the existing operating procedures followed by the involved technology partners
- Definition of vision scenarios
- Definition of a wide range of use cases, through which the functionalities of the CPSwarm workbench will provide a solid connection for the implementation of the vision scenarios.
- Definition of final use cases for different types of identified actors
- Capture, analyse and communicate end-user needs for the proposed technology in an effective manner through the process of drafting the use cases.
- Determination of technological components and services to be used

The results of Deliverable D2.1 have been continuously updated and refined through an iterative process that will lead to the production of this deliverable. The development of this deliverable was coordinated by ROBOTNIK with contribution of DiGiSky, TTTech, FRAUNHOFER, KLU, ISMB, SOFTEAM, LAKE and SLab. The outcome of this deliverable along with the deliverable D2.1: *Initial Vision Scenarios and Use Case Definition*, and D2.3: *Initial Requirements Report* reflects the final results of the Task 2.1: *Vision scenarios, use cases and initial requirements*.

1.1 Related documents

ID	Title	Reference	Version	Date
D2.1	Initial Vision Scenarios and Use Case Definition			M4
D2.3	Initial Requirements Report			M6

2 Approach and Methodology

As depicted in Figure 1, the development cycle for the CPSwarm Workbench starts in the top left with a scenario thinking methodology accompanied by collecting other kinds of input such as related work, documents, standards or available technologies. The input is then condensed in a high-level analysis in order to identify a first set of requirements and innovations that need to be met or become available, respectively, from the scenarios derived. In long-term iterations, system design, integration of technologies and knowledge as libraries takes place that is then implemented in an incremental manner and validated. The results from the validation are then fed back into the scenarios and collection of available knowledge base. New findings, corrections and additions are then incorporated into the existing documents and requirements as well as ideas for innovations are updated. This way, the cycle starts again, affecting all technical work, which is, in the end, validated again. This methodology allows for step-wise knowledge acquisition and development allowing for adjustments alongside conception and development.

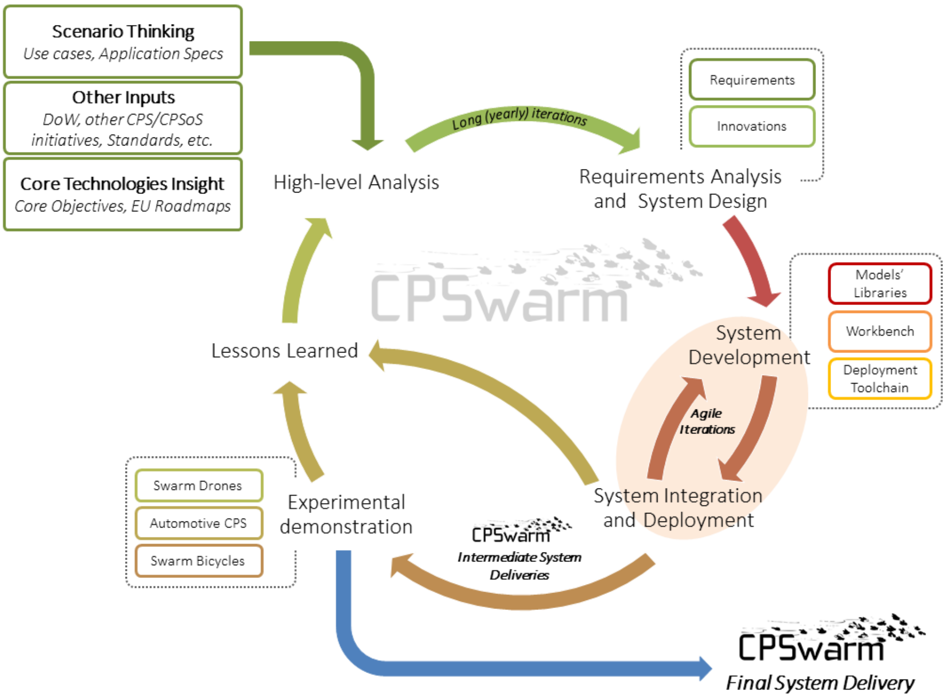


Figure 1: The CPSwarm Workbench development lifecycle.

The work reported in this deliverable is located in the top left corner and follows a user-centred approach for use case and, later in D2.3, requirements elicitation. The documents' structure reflects the different steps performed:

- Chapter 3: First, *final vision scenarios* were created for the different domains targeted in the CPSwarm project: (i) Surveillance and Emergency Response, (ii) Logistics and (iii) Automotive. The vision scenarios provide a futuristic picture for these domains and help thinking in visions of how things would work when CPSwarm workbench was available. This way, the project and research team receive a general orientation along which the efforts for conception, architecture design, components extension and integration as well as the scientific and general project communication can be aligned. Thus, the team has a common picture to discuss about and work on. Additionally, any discussion can be led alongside a common "tangible" scenario.
- Chapter 4: Second, from the final vision scenarios, all partners needed to understand on an abstract level how things work with currently existing technologies and processes and how a vision scenario could roughly be realized. This is performed by conducting technology demonstrations by the three industry partners of CPSwarm namely, DiGiSky, Robotnik and TTTech.

- Chapter 5: After gaining insight on the tools of the trade and the processes followed, the next step was to conduct brainstorming sessions to define the scope of the problem to be dealt with in the entire course of this project. As a result, a common understanding was created with regard to the modes of operation of the various components and handover point for integrating them within the CPSwarm workbench concept. In addition, gaps were identified concerning the flow of information as well as data formats and needs for defining harmonious flows for the workbench interaction.
- Chapter 6: The final step was to elicit use cases from the outcome of the brainstorming sessions on scenarios and thee demonstrations of processes and technologies. Roles, actors, stakeholders and action performed during the current development processes for drone scenarios were extracted. Sometimes, names had to be found for actors and roles since they are currently not defined explicitly. Their differentiation is, however, essential for structuring the CPSwarm workbench interaction flow.

3 Final Vision Scenarios

The following sections describe the current version of the scenarios, including the iterative process of revision, extension, refinement and maturation process of the set of use cases.

3.1 Swarm Drones

In CPSwarm, we will consider heterogeneous swarms of ground rovers and UAVs to conduct certain missions in Search and Rescue (SAR) tasks.

Final Vision Scenario

An explosion at the power plant in Pripyat has caused an electric black out in the entire town. There is fire everywhere at the plant. Workers are stuck in the middle of fire. Fire fighter vehicles have arrived. They are trying to control the fire. It is difficult to estimate how many casualties are still in danger. The captain of the firefighter's team decides to survey the whole area for casualties. He goes to the control station and feeds in the details of the power plant and fire coverage into the CPSwarm surveillance framework. The framework analysis shows that there is a need to send in a swarm of 8 drones and 4 rovers to provide full coverage. The swarm goes over the area, collaborating with each other to identify the casualties as well as the fire-free path for the members of the swarm. The drone flying over the shop floor identifies that there are 2 casualties stuck in the shop floor. It sends the signal back to the control station and help is sent to the identified location.

The swarm's mission itself is defined in a central operation station in the beginning of the mission with a dedicated swarm definition tool (mission planner) that defines the goals and behaviour of the swarm, thanks to the CPSwarm Deployment Toolchain. The central station can additionally collect the sensor data and perform sensor fusion and analysis in real time. The vehicles can also form a meshed ad-hoc communication network to improve communication performance. The central station is equipped with suitable user interfaces to enable the operator to influence the swarm (e.g. tell that he wants to see a certain scene then the swarm automatically sends the closest vehicle to the scene or to document the swarm mission including all GPS and sensor data.

Figure 2: Surveillance planning within the Swarm Drones scenario.

The gathered information is used to help security personnel, first responders as well as rescue teams to conduct their mission efficiently. The two application scenarios share common requirements such as:

- A vast spatial area has to be inspected and information has to be provided to the stakeholders (security personnel, rescue teams, etc.) in real-time, especially in case of an incident.

- Swarms need to reduce the inspection/detection times compared to, e.g., single UAV/rover applications.
- The inspection cycle time for surveillance needs to be reduced considerably enabling denser inspection. Especially in SAR, a single minute can decide between death and life.
- At the core, there are multiple UAVs and multiple rovers that can act autonomously. They need to carry different sensors (VIS or IR cameras, microphones, gas sensors, etc.). Either each vehicle carries all sensor modalities or the sensor modalities are distributed among the vehicles.
- The vehicles carry intelligence and can communicate among each other (via WiFi, 4G or others). They should operate as a self-organizing mixed team where particular tasks for each vehicle are not predefined at mission start but negotiated during mission execution.
- Such a swarm needs to be highly adaptive to changes in the environment and can act dynamically. For example, a ground robot may order a camera UAV to look for the best path, or a UAV finds something strange and orders UAVs with other sensors to check or asks if one of the rovers can move there to perform some action.
- Moreover, in contrast to fully centralized control, such a swarm still needs to be able to operate even if the connectivity among vehicles or with a base station is sparse.

3.2 Autonomous Freight Vehicles

For this scenario, the vehicle platooning concept is addressed combined with swarm behaviour and evolutionary algorithms.

Vehicle Platooning Concept:

- The leading vehicle has autonomous driving capability and prescribes the actions and decisions (i.e. navigation, decision on take-over manoeuvres, sequencing manoeuvres, lane change etc.) for the follow-up vehicles.
- The follow-up vehicles have autonomous driving capability and environmental awareness. They follow the leading vehicle´s actions.

Figure 3: Vehicles in platooning configuration.

Freight vehicle platooning holds great potential to make road transport safer, cleaner and more efficient in the future. Platooning results in a lower fuel consumption, as the trucks drive closer together at a constant speed, with less braking and accelerating. Freight vehicle platooning has also the potential to reduce CO_2 emissions. Likewise, connected driving can help improve safety, as braking is automatic with virtually zero reaction time compared to human braking. Finally, platooning also optimises transport by using roads more effectively, helping deliver goods faster and reducing traffic jams.

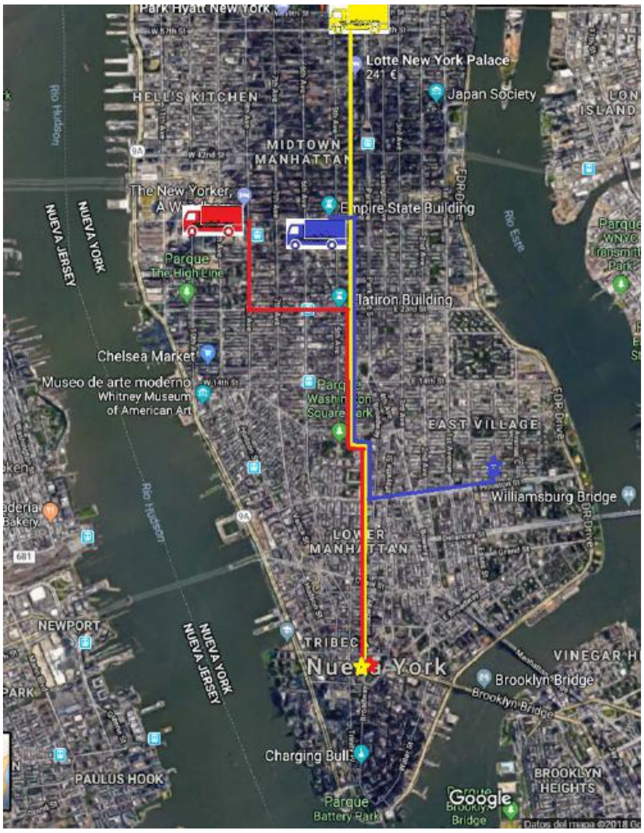


Figure 3-1 Example of 3 vehicles in a platoon

The yellow vehicle is an autonomously driving passenger transport vehicle that takes a group of tourists at The Hotel Plaza that want to visit the Tribute Museum. The red vehicle is an autonomously driving special goods transport vehicles and is at the Pennsylvania Train Station where it has picked up a new sculpture to bring it to the Tribute Museum. The blue vehicle is also an autonomously driving passenger transport vehicle that has picked up a group of travellers that have already visited the Empire State Building and want to go now to the Neyorican Poets Cafe located on the East Village, where they will have some rest.

Since a certain part of the route is common for the three vehicles, they decide to create a platoon. The blue vehicle joins the yellow vehicle as a follower whereas the yellow one leads the platoon. When they arrive at the Flatiron Building, the red vehicle joins them as a second follower vehicle. The three vehicles run together until Houston street where the blue vehicle leaves the platoon to go to its final destination. The yellow and the red vehicles keep the platoon until the Tribute Museum where they both reach their final destination.

Mission

The mission describes the goal of the vehicle, i.e., where it currently is and where it should go. The optimization requested will be done on the route needed to execute the mission with the least cost. When two or more vehicles are traveling behind each other the cost of the road is reduced by 20% making platooning as preferred solution. The goal of the optimization will be to find out the best route for every vehicle.

Additionally, the vehicles have full autonomous driving capability thus they are able to take decisions when, for example, an obstacle on the route does not allow them to continue with their route. In such a case they can change the lane (drive at left lane behaviour) or brake to a full stop (emergency braking behaviour).

Other behaviours will be:

- the shortest path algorithm for each vehicle, from start position to final destination, responding to evolved or swarm algorithm provided by the CPSwarm workbench.
- Join/leave the platoon, responding to evolved or predictive algorithm provided by the CPSwarm workbench.

Some vehicles might create a platoon while others will only follow the shortest path as fully autonomous vehicles. The vehicles that will create the platoon in the common route, will select their role (either leading vehicle or following vehicle) dynamically based on evolutionary algorithms.

On the other hand, when they create a platoon, due to the small distance among vehicles, some of the vehicles' sensors might become impaired (e.g. from camera); which means that they can only rely on the leading vehicle. The vehicles are connected via deterministic wireless (explained later) when they run in a platoon so that the leading vehicle can prescribe actions and decisions (i.e. navigation, decision on take-over manoeuvres, sequencing manoeuvres, lane change etc.) for the follow-up vehicles. Relevant properties of such a distributed automotive system are modelled with the CPSwarm workbench modelling tool.

3.3 Swarm Logistics Assistant

The SWARM Logistics Assistant scenario involves robots, rovers and drones that collaboratively perform opportunistic scanning of the warehouse. The idea is to scan the entire area of the warehouse and share the acquired information to update the knowledge base on the go. In addition to collecting information about the maps of the entire area, the connected robots will also be used for collecting additional information implicitly e.g. room temperature, presence of humans, detection of in-path obstacles etc. Since all the connected robots of the swarm acquire the information collaboratively, the status of the area is always up to date and the effort is always divided among all members. As a starting point, each connected robot will be fed with some default information e.g. map of the area. This information is updated opportunistically on the go as the robots perform their main tasks. The main tasks of the robots are intended to assist human in a logistics domain. These assistive tasks could include joining forces to move a heavy obstacle from one place to another.

Figure 6: Impression on the Swarm Logistic scenario.

Additional sensors could be used to attain information regarding various aspects e.g. accelerometer for floor conditions, thermal sensor can detect the presence of any humans/animals in the vicinity, gas sensors can detect leakage of any harmful gases, temperature sensors can extract information about the current temperature of the entire area. For example, a warehouse is required to maintain a certain temperature to preserve the quality of fruits stored in it. While the robots are moving a fruit package from one point to another, they opportunistically detect the temperature of the area and in case, the temperature is higher/lower than the desired value, the respective personnel will be notified.

Final Vision Scenario

Everyday a truck comes into a warehouse with a lot of packages that must be stored. The work of Alex is to put all the packages in tagged carts.

Figure 7: Alex needs to put the packages in tagged carts.

There is a storage area inside a warehouse where carts with ware staff are stored.

These tagged carts must be transported to a specific storage area dedicated to this. Alex always try to do it occupying the minimum possible space to optimize the storage space into the warehouse.

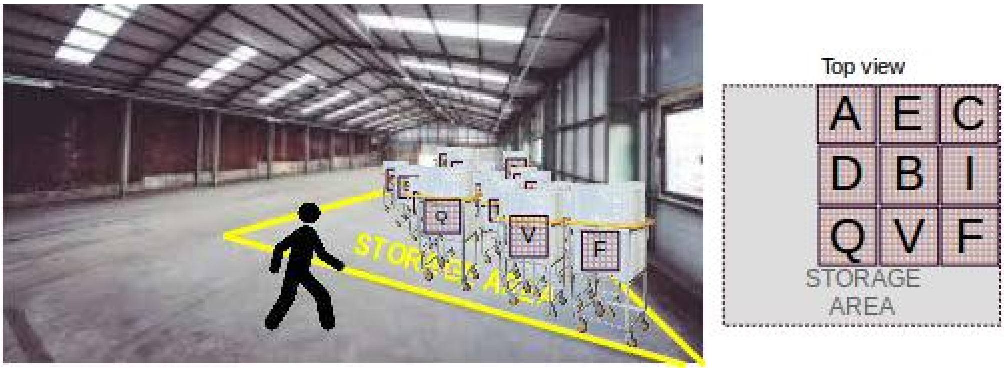
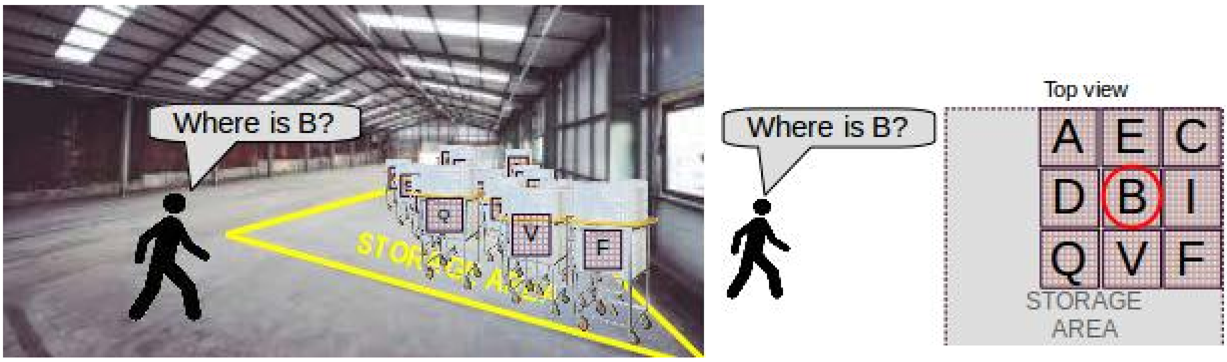


Figure 8: Dedicated storage area to store all the carts.

Periodically a specific cart is requested and Alex must look for it inside the area and transport it to another place.



67

Figure 9: Alex needs to pick the cart B.

Usually the requested cart is surrounded by other carts so Alex has to withdraw some carts to get it, spending too much time in this operation. In addition, usually more than one cart is requested at the same time so sometimes there are more than one operator (Alex's colleague) inside the storage area trying to get each one his cart.

By other hand, new carts come into the warehouse and they must be stored in the same area so at the end there is a lot of people surrounded the area where the carts are stored without any kind of organization.

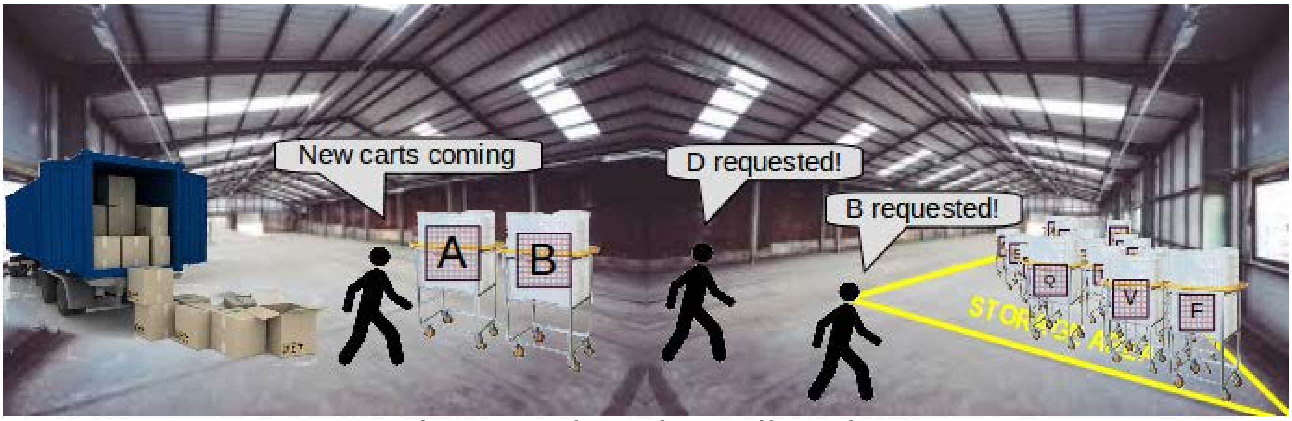


Figure 10: Workers trying to self-organize.

For this reason, the company installs a CPSwarm system to manage the storage area automatically and organize carts in an optimal way. The system is responsible of organizing the carts inside the warehouse and preparing the requested carts in a manual load area. This saves Alex a lot of work with regard to finding an optimal solution that needs to total overview of all stored carts as well as moving them. The system finds the optimal solution and the autonomously acting rovers move new or demanded carts from or into the manual load area where Alex can further handle them.

The robots recognize the legs of the carts through the measures from a laser scan and they will move under the selected cart. With an integrated an elevator above the upper plate, the robots are able to lift, move and release carts as shown by the following image.

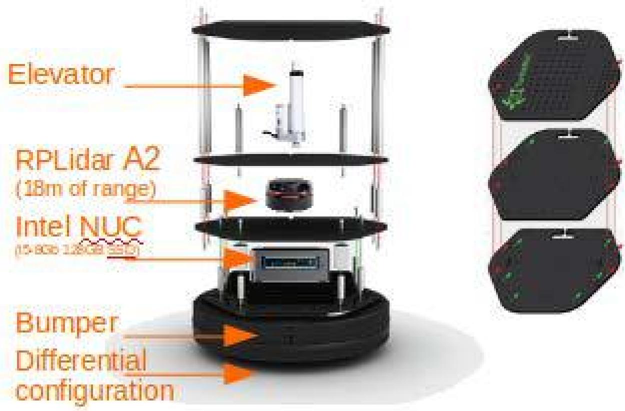


Figure 11: New structure of the robot.

The robots with the CPSwarm system must try to optimize the space occupied by the carts inside the warehouse and also try to serve the requested carts as soon as possible when they are required.

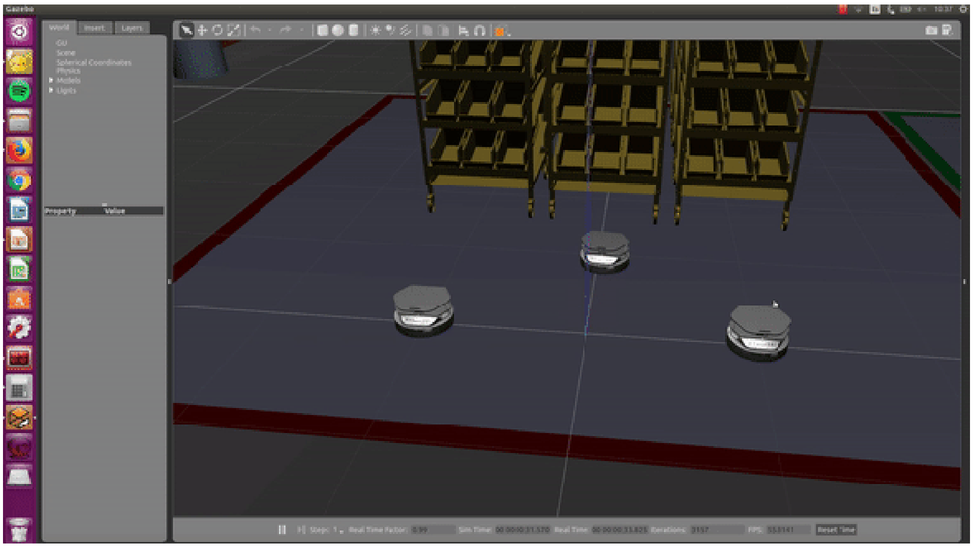


Figure 12: Gazebo simulation for the three turtlebots and the carts.

Alex is very happy with his job now because he knows that the CPSwarm system helps him saving time and lowering the physical efforts for every new order.

4 Technology Specification

The technology exploration served as foundation for the ideation phase described in Chapter 5 where a working scenario is given along which the different steps to be performed with the CPSwarm workbench are followed. The following sections describe the domain analysis performed for the three domains covered in the scope of the CPSwarm project. The first step towards understanding the domain of realizing swarm applications was to conduct technology demonstrations. Currently used technologies and tools used by the consortium as well as the ones suited for the purpose of the project were elaborated by semi-structured interviews conducted remotely with the CPSwarm partners. For the first project iteration, the efforts concentrated on assessing available technologies from the partners DigiSky, TTTech and Robotnik. In addition, the Modelio UML tool suite (SOFTEAM) was analyzed together with the available technologies for optimizing swarm algorithms brought into the consortium by the partners Lakeside and UNIKLU. For the scope of this deliverable, the currently applied tools and processes from the partners DigiSky, TTTech and Robotnik are presented. The results of the other interviews will be used for technology review in D3.1.

Each demonstration consisted of a brief introduction to the technologies in use and different processes being followed for development and deployment purposes. The sessions were held interactively and questions were allowed by all participating partners at all times. This way, a dialogue between all involved teams could be established. During the interview sessions, the following subjects were covered each time:

- Currently established tools and processes by the partner
- Data / Information handover points between the different components of the CPSwarm workbench
- Research focus within the CPSwarm project
- Current limitations and a "wish list" on what features a workbench should provide

4.1 DigiSky - Development for UAVs

The technology demonstration prepared by DigiSky explained the currently used tools for developing, simulating and deploying applications to unmanned aerial vehicles (UAV) - aka "drones". The results are summarized in the following:

- Systems for drones can be differentiated between a very responsive flight control / auto-piloting system that is predominantly designed for the process of controlling vehicle in real-time.
- The auto-piloting system is based on PixHawk (hardware) and PX4 Flight Stack (software).
- The drones' operating system is called NuttX and makes use of the PX4 Driver Framework. Latter represents a middleware layer similar to ROS such that new devices only need to be connected to the middleware and application code can remain unchanged.
- The uORB Messaging Mechanism is similar to the one used be ROS for pub/sub topics.
- Communication with external devices is based on MAVLINK
- In this regard, technologies should be similar to the approach of ROBOTNIK since the same auto-piloting and control components are used.
- Additional microcomputers can be augmented with different kinds of external hardware such as sensors but do not immensely influence the energy consumption of limit the payload of a drone.

W.r.t. the communication capabilities:

- The maximum distance for data communication between ground station and drone lies up to 10km.
- The maximum distance for video transmissions is up to 1.7km.

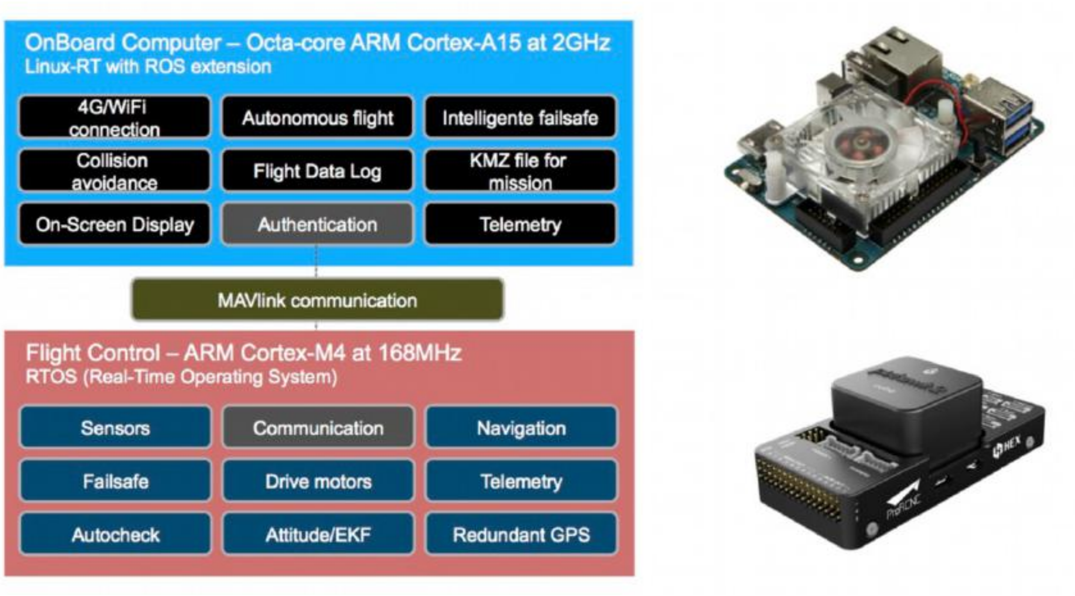


Figure 13: DigiSky's UAV architecture.

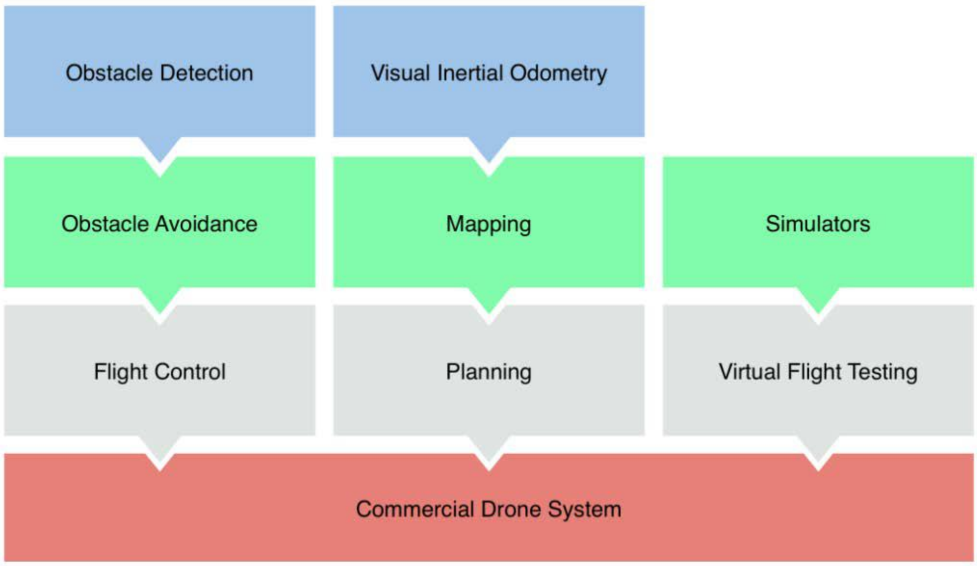


Figure 14: Overview of flight control and auto-piloting features.

4.1.1 Used Tools and Applied Processes

Open Source tools collection and source code based on work originally done at ETH Zurich.

- Programming is mostly based on C++ that allows easy application and driver development. Easy to add new models
- Code is built with any C++ IDE and compiled by following the usual deployment chain (e.g. cmake)
- Gazebo simulates the code models and is based on ROS. The code for simulation is the same one as the one finally deployed on drones
- Simulating several drones is possible
- A flight simulator is used for testing developed code.
- Deployment is currently performed on ground via USB connection (rotors need to be removed for safety reasons).

- Missions can be updated during flight.
    - Mission, e.g.:
        - Behaviour
        - Waypoints
    - Updates are in general possible if models do not change

4.1.2 Handover - Points

- From models to code / simulation
- Making use of the same code for simulation and deployment

4.1.3 Focus within CPSwarm project

- CPSwarm Workbench
    - Conversion from rules / models / modules to the flightstack software
        - model --*conversion*-->compiled version --*semi-automated conversion* --> flightstack
        - This raises questions for the deployment toolchain
- Focus on application scenarios: Surveillance and emergency response and deeper analyze feasibility
- Learn more about swarm algorithms
    - Eventually directly implement some aspects into the auto-piloting software

4.1.4 Wish List

- MAVROS could be used as abstractions layer, e.g. for drones (DigiSky) and rovers (ROBOTNIK) such that one code version can be used for different devices. Functions that cannot be used on the device are neglected. Simple example: Implement moving directions once --> Translation for each device individually.
- Additional functionality should be added by additional hardware, e.g. ARM based microcomputers. Application ideas could be:
    - Image recognition
    - Recognition of moving objects
    - Human recognition
    - Sensing
    - More intensive computations

4.1.5 Further Reading

- Pixhawk [2]
- PX4 Development Guide (Wiki) [3]

4.2 TTTech - Networking Technologies

The automotive use case will be a laboratory level demonstrator (TRL 3 to TRL 4, demonstration in broadboard lab environment) around autonomous driving vehicles connecting them via a kind of electronic drawbar. Since the electronic platforms are mainly used for intra-vehicle computation, the aim of the demo is to allow inter-vehicle communication to enable coordinated actions such as the ones described in the scenario. To do this, a wireless connection among the vehicle computing platforms should be established.

The demo will be composed of:

- Exemplary communication systems of 3 vehicles are built in the lab demonstrator: 1PC + 2 fog nodes with distributed ECUs
- The demo might be complemented with more simulated vehicles, if necessary
- Environmental awareness will be simulated for all of them using a simulator feeding live data from simulated sensors

4.2.1 Used Tools and Applied Processes

The laboratory demonstrator consists of a central computer, (1) an automotive fog node, which is a virtualized high-performance computer, envisioned for the future car architectures, and (2) automotive electronic control unit (ECU). The MFN 100 fog-node has a building switch.

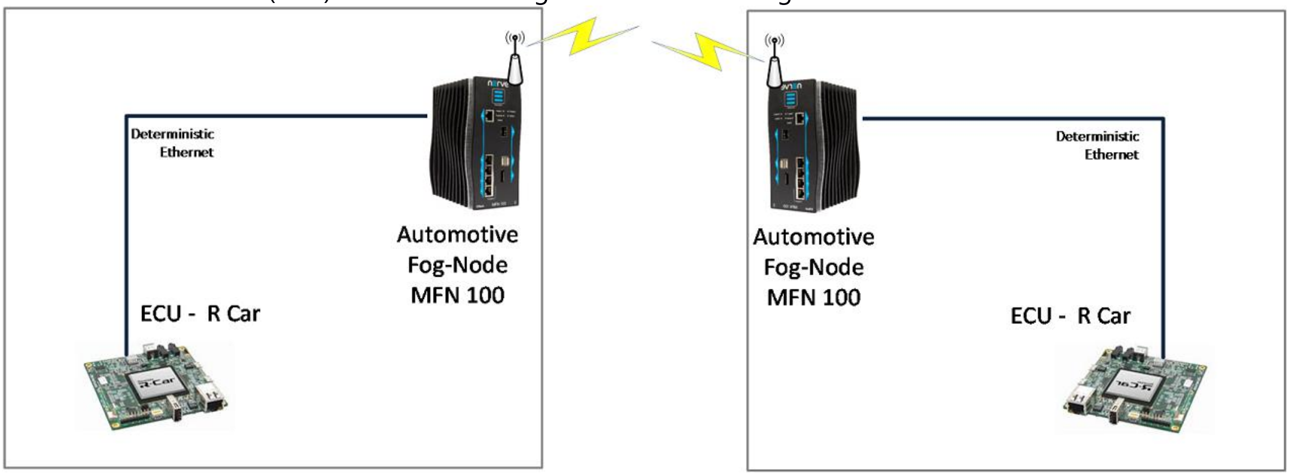


Figure 15: TTTech's Fog node in the lab demonstrator

The idea of the fog node is to bring some computational power and data processing closer in the vehicle and "offer" this real-time processing power to other units in the vehicle, with rather restricted computing power. Data to be sent to the cloud can be also pre-processed so that the transmitted data is reduced. The fog node acts thus also as a gateway between the Cloud and other end devices (ECU).

The proposed architecture is:

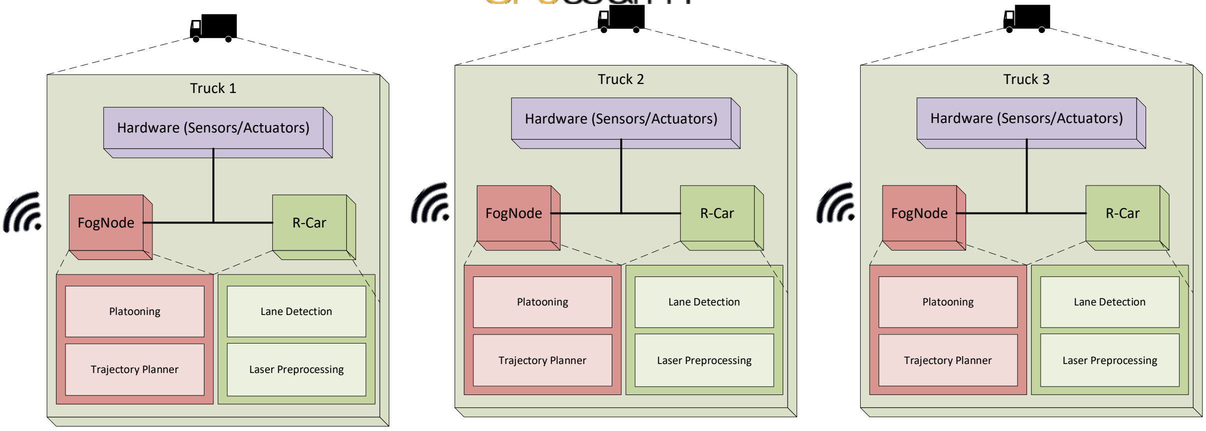


Figure 4-1 Architecture of the automotive use case

Each of the vehicles is considered a black box and is responsible for each own sensors and actuators. Only the FogNode is visible from outside (any communication from externals goes through it).

The vehicles, through the FogNodes, can communicate with each other via wireless. They can exchange information such as the speed of the vehicle, exceptional situations (e.g., emergency breaking), the fuel level or any other information crucial to the mission.

Safety in Automotive

In Automotive, assuring safety properties and proving compliance with safety standards counts from the very beginning when developing safety-critical embedded systems. One of the aims of the CPSwarm research is the technical implementation of a kind of drawbar over wireless connection. For the vehicles to react appropriately to the information from other vehicles, the information must arrive in a reliable and timely predictable way (deterministic real-time). Therefore, the challenges for the data communication among vehicles are:

1. Wireless communication

The communication from the leading vehicle to the follower vehicles, and also among all the vehicles, must be mandatorily wireless since it is not possible to have a wire among vehicles when they are running in a realistic situation.

2. Real-Time communication

Real-time communication is compulsory to give response to the safety requirements, for example, when breaking. Network communication technology must use time scheduling to bring deterministic real-time communication.

3. Low reliability communication

Real circumstances like harsh weather conditions, obstacles or presence of other wireless signals may decrease the reliability of wireless transmissions and can compromise real-time communication requirements. Considering that the quality of the wireless channels varies with the time, frequency and location, it is possible to increase reliability by finding better times, frequencies and locations to transmit and/or by performing retransmissions, while still observing deadlines.

To overcome these challenges, a wireless driver that meets all the requirements of the car convoy scenario is under development. The driver is based on the IEEE 802.11 standard for local wireless area networks.

OMNeT++ discrete event simulator will allow testing whether the communication between vehicles will perform in realistic environments with heavy vehicle traffic and wireless channel conditions as found in roads or city streets.

4.2.2 Handover - Points

- Drive at lane X
- Emergency Breaking
- Change Cruising Speed
- Change Max Speed
- Shortest path algorithm
- Dynamic selection of the role (leading/follower vehicle)

4.2.3 Wish List

- Deterministic wireless networking
- Electronic drawbar application

4.3 Robotnik - Customization and Development for Rovers

During the technology demonstration, Robotnik introduced their research focus and workflow.
Robotnik mainly focuses on mobile robots. Hardware and software specifications of robots are adjusted according to the needs of current projects.

4.3.1 Used Tools and Applied Processes

ROBOTNIK therefore analyses the usage contexts and suggests fitting robot models as well as suited pieces of the software frameworks that are then customized with regard to current needs. Thus, for projects, there are phases for:

- Analysis
- Modelling and choosing available assets
- Customization and development
- Simulation and testing
- Deployment and operation

Currently used tools:

- Software is developed based on Ubuntu with the ROS (Robot Operating System) Framework
- Solid3D models can be integrated into the process The primary languages to be used are C++ and Python
- Gazebo and AGVS simulation environments. Simulation is used as much as possible to lower the time needed for testing with real robot.
- Rviz [12] from ROS is used as the visualization tool for robots

In ROS, the structure of a robot is defined in a ROS standard format called **URDF**[1] file. The URDF file is based on XML text. Right now, there is no way to automate the process of writing the URDF file. Developers have to edit XML files manually. This could be one point where we can improve with Modelio [13].

Gazebo is used as the simulator for robots. Gazebo has very good integration with ROS and it is possible to simulate multiple robots. Gazebo utilizes physical engine to simulate the real world environment. Gazebo uses SDF (Spatial Data Format) files for building the simulated environment, which is also based on XML. This could also be another point to use Modelio.

1 URDF : **http://wiki.ros.org/urdf**

In AGVS, a situational picture can be created on what kinds of sensor data the robots receive. Changes in the Gazebo environment are visualized in AGVS.

ROBOTNIK puts lots of efforts in customizing / programming and simulation / deployment due to a very large variety of parameters regarding the robots' and algorithms parameters. Depending on the usage, scenario, a suited configuration needs to be found for every situation. This kind of fine-tuning is a very time-consuming task.

For CPSwarm, the robot model Turtlebot 2[2] will be used as member of swarm. A Celeron single-core Linux PC with ROS drives the robot. The turtlebot provides Wi-Fi access so it is possible to communicate with the centre or with other turtlebot in a swarm. Swarm algorithms could be deployed as ROS node on the Linux PC. The deployment could be done using SSH through Wi-Fi. Extensions on the hardware configuration like multiple CPUs to increase computational power are possible.

Several programs can be deployed on the robot's OS such that a switch can be triggered manually. Regarding behavioural change, it is also possible to provide program logic that switches based on contextual information or other triggers.

4.3.2 Handover - Points

- From models to code / simulation
- From configuration to simulation

4.3.3 Focus within CPSwarm project

- Improve multi-robot use case
- Improve ROS interface with other systems
- Address cross-platform development --> become more independent from ROS
- Replace the traditional central control architecture with swarm behaviour
- Find ways to define update channels for configurations / program logic
- Use CPSwarm workbench to reduce commissioning time
- Improve the process of configuring / testing robots and algorithms

During the conference, an EU project called RAWFIE [http://www.rawfie.eu/] was also mentioned as an example of connecting robots from Robotnik with other vehicles, e.g. drones.

Some further interesting readings are the next:
- The ROS Framework [4]
- Gazebo5 [5]
- The RAWFIE Project [6]
- ROS Components [7]
- GitHub Repository (Robotnik Automation) [8]

2 http://www.robotnik.eu/mobile-robots/turtlebot-ros/

5 Use Case Analysis

5.1 CPSwarm Workbench Context of Use and Workflow

The CPSwarm workbench aims at easing the process of engineering deployable, tested solutions. The focus is not centred on creating ad hoc solutions for the need of the hour. On the contrary; the purpose of the CPSwarm workbench is to fully design, validate and deploy engineered solutions. The CPSwarm Workbench is meant to design swarms consisting of drones, rovers and/or other kinds of robots. As a result of the brainstorming session, the following workflow was designed for the CPSwarm workbench (Figure 15):

- First, the treated problem needs to be engineered and understood before the problem itself can be designed. In a follow-up step, modelling of the problem, the environment, strategy to find a solution and used hardware takes place.

- The output of the modelling phase is used as basis for optimizing and refining the used swarm algorithms that make use of all information available via the specified models.

- The next step consists of running simulations on the derived algorithmic solution and to benchmark the solution. It may become necessary to iteratively refine the models or change parameters of the used algorithms before reaching a proven solution.

- The accepted solution that is verified by simulations is then deployed and run on swarm devices. Tests and usage in real settings are then used to verify and validate the derived solution.

- In case modifications are needed, all steps can selectively be repeated or modified, such that simulation and deployment are used to validate the modifications again.

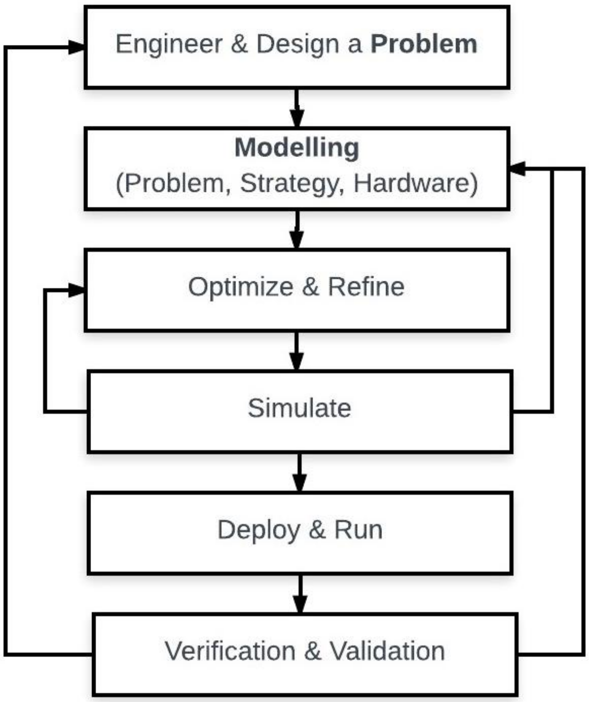


Figure 18: CPSwarm workbench workflow.

5.2 Identified Stakeholders and Relevant Actors

In software and systems engineering, a use case is a list of steps [10], typically defining interactions between the actors and the system in order to achieve a goal. The involved actors can be human or an external system. Usually use cases represent tasks or stakeholder goals. A stakeholder is a role played by a person, place or thing that has some sort of interest in the outcome of the processes and activities. Stakeholders are not to be confused with people, since stakeholders are not persons in an organization but rather roles that a person can have. In many cases, a person can be assigned with more than one role, while also a role can be associated with more than one person. As a result, a stakeholder is a role that a person, the environment, a place or a thing can have.

In the following identified stakeholders and categories to which they belong are listed. Not all identified roles are relevant for the CPSwarm Workbench design, therefore those marked by as **bold** are further consider in the use case analysis. Within the use-case descriptions, these roles will be referred to as *actors* following the UML terminology since they are interacting with a system.

In total, three categories were derived that are related to the interest in the CPSwarm Workbench:

- **Commercial**: Stakeholder of this category have a business-related interested in the CPSwarm Workbench.

Table 1 - Commercial stakeholders.

Stakeholder	Description
Customer	A person, group or an organization interested in the purchase of the workbench
Workbench Engineer	A person, group or an organization responsible for the development and maintenance of the workbench

- **Design**: This group makes use of the CPSwarm Workbench in order to fulfil their design and implementations tasks in order to achieve an application goal.

Table 2 - Design-related stakeholders.

Stakeholder	Description
Mission Planner	A person responsible for planning the mission. The mission includes: • Problem definition • Approach to solve the problem • Environment description • Mission parameters • Mission success condition
Swarm Designer	A person responsible for designing the swarm based on the mission defined by the mission planner. The swarm designer analyses the given problem and designs the structure and behaviour of the swarm and its single members accordingly.
Domain Expert	A person, group or an organization who is an expert of the problem domain. He is responsible for providing expert advice about the domain e.g. rules, regulations, limitations etc.
Swarm Modeller	A person who constructs the model of the swarm and its members. This model is the visual representation of the structure and behaviour of the swarm specified by the swarm designer.
System Architect	A person who is responsible for defining the architecture of the internal implementation of the swarm. He ensures the coordination between the swarm modeller, developer and the algorithm & simulation expert.
Swarm Developer	A person or a group responsible for adding logic to the generated code. This code is later on deployed on each component of the swarm.
Algorithm Optimization and Simulation Expert	A person or group who provides the expertise regarding the swarm algorithm. He decides the aptness of a certain algorithm given a specific swarm problem.
Safety and Security Expert	A person, group or an organization responsible for providing expertise on safety and security of the swarm.

Deployer	A person or group responsible for deploying the code of the swarm.

- **Operation**: These stakeholders are interested in the swarm behaviour, control and feedback at runtime.

Table 3 - Operation-related stakeholders.

Stakeholder	Description
Data Analyst	A person or group responsible for viewing and analysing the data acquired, both actively and passively, by the swarm in real-time. He assists the decision maker by interpreting the acquired data in a meaningful way.
Decision Maker	A person or group responsible for calling all the shots. He relies heavily on the information received from the data analyst to make operational decisions about the swarm in real-time.
Swarm Operator	A person with the command control in his hand. He is responsible for directly manipulating the components of the swarm.
System Hackers	A person, group or an organization with the intention to breach the system.
Affected Humans	A person or group who interacts with or is assisted by the swarm in order to complete a mission.

5.2.1 Communication Flow between Stakeholders

The previous section explained various stakeholders and their responsibilities. Figure 6 shows the communication flow between all the identified stakeholders. The flow starts with the customer who defines the need of the swarm to solve a problem in a given situation. The problem and the details of that particular situation is then communicated to the mission planner. The mission planner is responsible for defining the problem, environment and the desired solution. He coordinates with the swarm designer and domain expert to formulate a swarm design as a solution to the predefined problem.

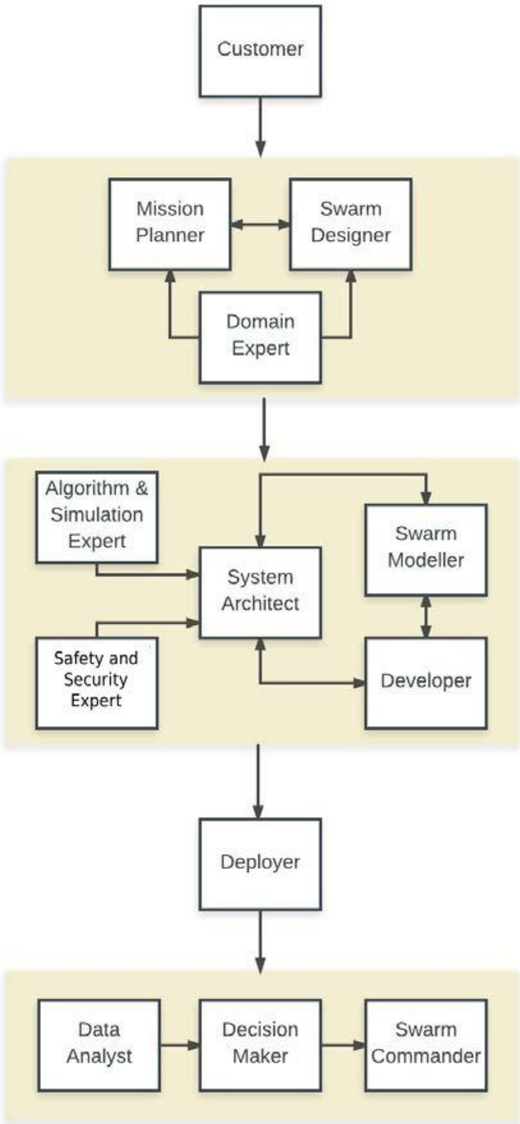


Figure 19: Communication flow between stakeholders.

After this design phase is concluded, this information is passed on to the implementation phase in which the system architect plays an extremely vital role. He is responsible for coordinating with the swarm modeller and developer to make sure that the design of the Cyber Physical Systems involved in the swarm as well as the design of the warm itself constructed in the design phase are correctly modelled by the swarm modeller and later on, accurately implemented by the developer(s). The system architect also communicates with the algorithm & simulation expert and the safety and security expert to gain expert opinion regarding the optimization of the swarm algorithm and security measures respectively. Once the solution is implemented, the code is passed on to the deployer, who is responsible for deploying the code on each component of the swarm.

Once the swarm is up and running, there is a lot of data acquired by the swarm both actively and passively. The data analyst(s) is responsible for analysing and interpreting this data into meaningful and understandable information. This information is passed on to the decision maker, who, as the name suggests, takes decisions as need be. Lastly, these decisions are enforced by the swarm commander/operator who is responsible for directly manipulating the swarm.

5.3 CPSwarm Use Cases

The following section contains all the identified use cases for the CPSwarm workbench. These use cases are organized with respect to the involved actors. The templates shown in Figure 7 will be used throughout this document to visualize an actor's actions with the system.

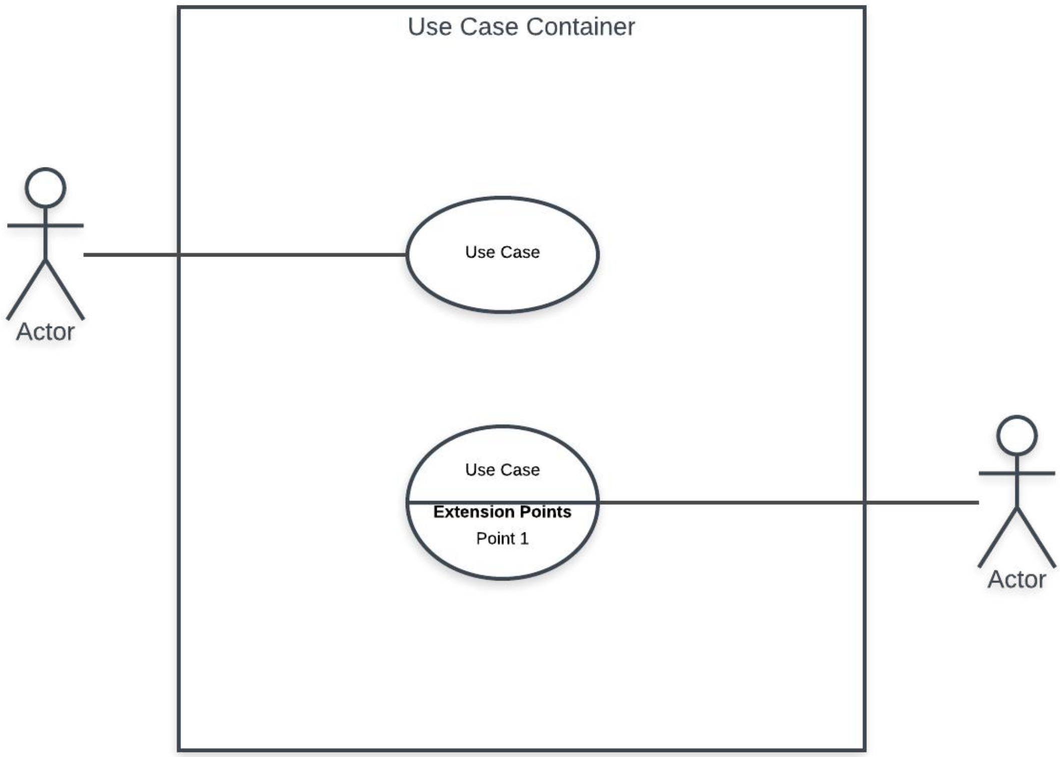


Figure 20: Use case diagram template

Each use case is then described in detail, following the template shown in Table 4.

Table 4 - Use Case description template

Use Case ID	Internal ID for use case identification
Use Case Name	Use case name describes the targeted action
Version	Stage the use case has reached
Author	Who documented the use case
Use Case Diagram(s)	References of related use case diagrams
Involved Actors	Actors involved in the use case i.e. people or system(s) who/which directly interact with the system
Preconditions	Preconditions specify the conditions that must hold true before the scenario of the use case starts

Trigger	What triggers the execution of the use case
Brief Description	A brief description of the use case
Post-conditions	Post-conditions specify what must be achieved at the end of a successful use case

5.3.1 Workbench Engineer

As described in Table 2, Workbench Engineer is a person, group or an organization responsible for the development and maintenance of the workbench. Figure 8 shows all the use cases related to the responsibilities of the Workbench Engineer, followed by their descriptions.

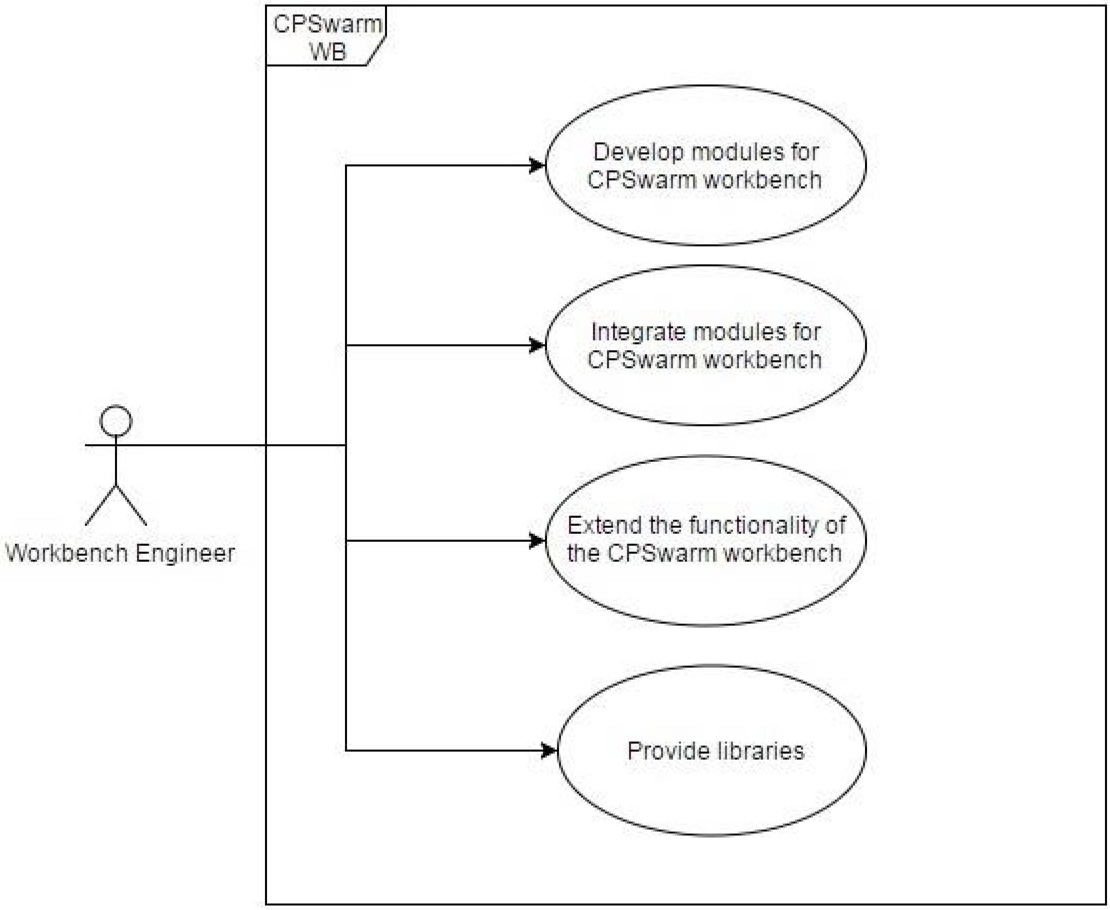


Figure 21: Workbench Engineer use cases.

Use Case ID	UC-10.1
Use Case Name	Develop modules for CPSwarm workbench
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 8
Involved Actors	Workbench Engineer
Preconditions	-
Trigger	Need for modules for CPSwarm workbench
Brief Description	This use case enables the involved actor(s) to develop a module for the CPSwarm workbench.
Post-conditions	Modules for CPSwarm workbench are successfully developed

Use Case ID	UC-10.2
Use Case Name	Integrate modules in CPSwarm workbench
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 8
Involved Actors	Workbench Engineer
Preconditions	Module is already developed
Trigger	Need to integrate modules in CPSwarm workbench
Brief Description	This use case enables the involved actor(s) to integrate a prebuilt module into the CPSwarm workbench.
Post-conditions	Modules are successfully integrated in the CPSwarm workbench

Use Case ID	UC-10.3
Use Case Name	Extend the functionality of the CPSwarm workbench
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 8
Involved Actors	Workbench Engineer
Preconditions	Basic functionality of CPSwarm workbench exists
Trigger	Need to extend the functionality of the CPSwarm workbench
Brief Description	This use case enables the involved actor(s) to extend the functionality of the CPSwarm workbench if need be.
Post-conditions	Functionality of the CPSwarm workbench is successfully extended

Use Case ID	UC-10.4
Use Case Name	Provide libraries
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 8
Involved Actors	Workbench Engineer
Preconditions	-
Trigger	Need for library
Brief Description	This use case enables the involved actor(s) to provide libraries for various purposes. An example can be a model library that can be used by the Swarm Modeller to create and re-use swarm models.
Post-conditions	Libraries are successfully provided

5.3.2 Mission Planner

The Mission Planner as mentioned in Table 2 - Design-related stakeholders, is responsible for formulating the description of the mission that is to be carried out by the swarm. This mission description includes the problem definition, approach to solve the problem, environment description, mission parameters and lastly, the success condition of the mission. Figure 99 shows all the use cases related to the responsibilities of the mission planner.

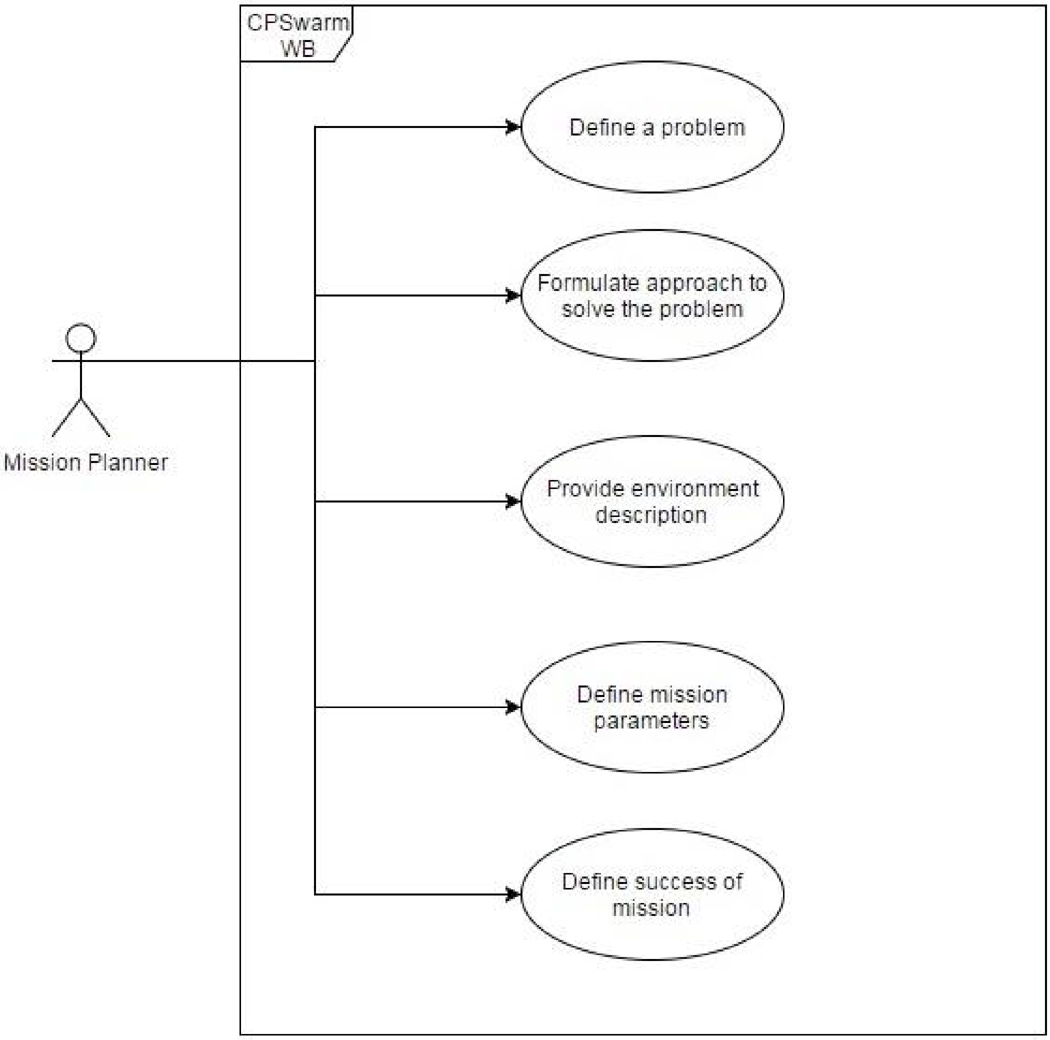


Figure 22: Mission Planner use cases.

Use Case ID	UC-1.1
Use Case Name	Define a problem
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 9
Involved Actors	Mission Planner
Preconditions	The need to create a mission
Trigger	The mission planner realizing a problem.
Brief Description	This use case enables the involved actors to define a problem. For example, in a power plant there is a need to survey the area for

	security purposes. This in itself is a problem. The mission planner takes this situation and extracts the problem out of it, which in this case is to *avoid external breach to ensure security of the area.* This is the starting point of the entire flow of communication between the mission planner, the swarm designer and the domain expert. The mission planner defines a problem, which is further communicated to the swarm designer who is responsible to allocate resources according to the scope and nature of the defined problem. The domain expert acts as an external check in this entire planning. He makes sure that all the external factors such as rules, regulations, constraints and limitations are kept into consideration.
Post-conditions	Problem is successfully defined.

Use Case ID	UC-1.2
Use Case Name	Formulate approach to solve the problem
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 9
Involved Actors	Mission Planner
Preconditions	A problem is already defined.
Trigger	The need to define the approach to solve the defined problem.
Brief Description	This use case refers to formulating an abstract approach towards solving a predefined problem. Considering a situation where there is a need to monitor and avoid any external breach in an area, a possible approach can be to *survey the entire area.* This approach to solve the problem, along with the predefined problem is shared with the swarm designer and domain expert for further analysis.
Post-conditions	An approach to solve the problem is successfully formulated.

Use Case ID	UC-1.3
Use Case Name	Provide environment description
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 9
Involved Actors	Mission Planner
Preconditions	A problem is already defined.
Trigger	This use case is triggered by the need to add environmental details to the predefined problem.
Brief Description	This use case enables the involved actor(s) to define the details of the environment for which the problem was defined. Area, height, wind direction, wind speed, weather conditions, traffic situation, landscape details and time of the day can be a few examples of the attributes that can help the mission planner in describing the environment in much detail. These attributes may also include the limitations that they impose. It is very vital to consider the details of the environment description before a concrete solution can be designed for a particular problem. This use case ensures that all the external factors that may influence the defined problem as well as the solution, are kept into consideration.
Post-conditions	Environment description is successfully provided.

Use Case ID	UC-1.4
Use Case Name	Define mission parameters
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 9
Involved Actors	Mission Planner
Preconditions	A problem is already defined.
Trigger	In case the mission requires any additional specifications.
Brief Description	Along with the problem definition, an approach to solve the problem and environment description, the mission may also include certain additional details such as time limit i.e. the solution has to be implemented within the defined time limit. This use case is used by the involved actor(s) to define any additional parameters or constraints in support to the predefined problem.
Post-conditions	Mission parameters are successfully defined.

Use Case ID	UC-1.5
Use Case Name	Define success of mission
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 9
Involved Actors	Mission Planner
Preconditions	Problem has already been defined
Trigger	This use case is triggered by the need to define the success of a specific mission.
Brief Description	This use case enables the involved actor(s) to define the success condition for a specific mission. In simple words, what state should be called as the state where the predefined problem is solved. It is very important to define the mission success condition concretely at the initial design stage so that this information can later on be used as a verification element to ensure that the mission is completed as intended. From the perspective of computation, this success condition can be later on translated into the fitness function for the purpose of optimization of the swarm algorithm.
Post-conditions	The success condition of the mission is successfully defined.

5.3.3 Swarm Designer

As described in Table 2, the swarm designer is the person who is aware of all the available resources and their capabilities. In the design phase, the mission planner communicates the mission details i.e. problem definition, solution approach, environment description, mission parameters and success condition of the mission to the swarm designer so that he can make an educated recommendation as to what and how many resources should be allocated to solve the problem at hand. For example, number of swarms needed, number of members in each swarm, type of member (drone, rover, …) etc. Figure 10 shows all the use cases related to the responsibilities of the swarm designer. Each use case is further described below.

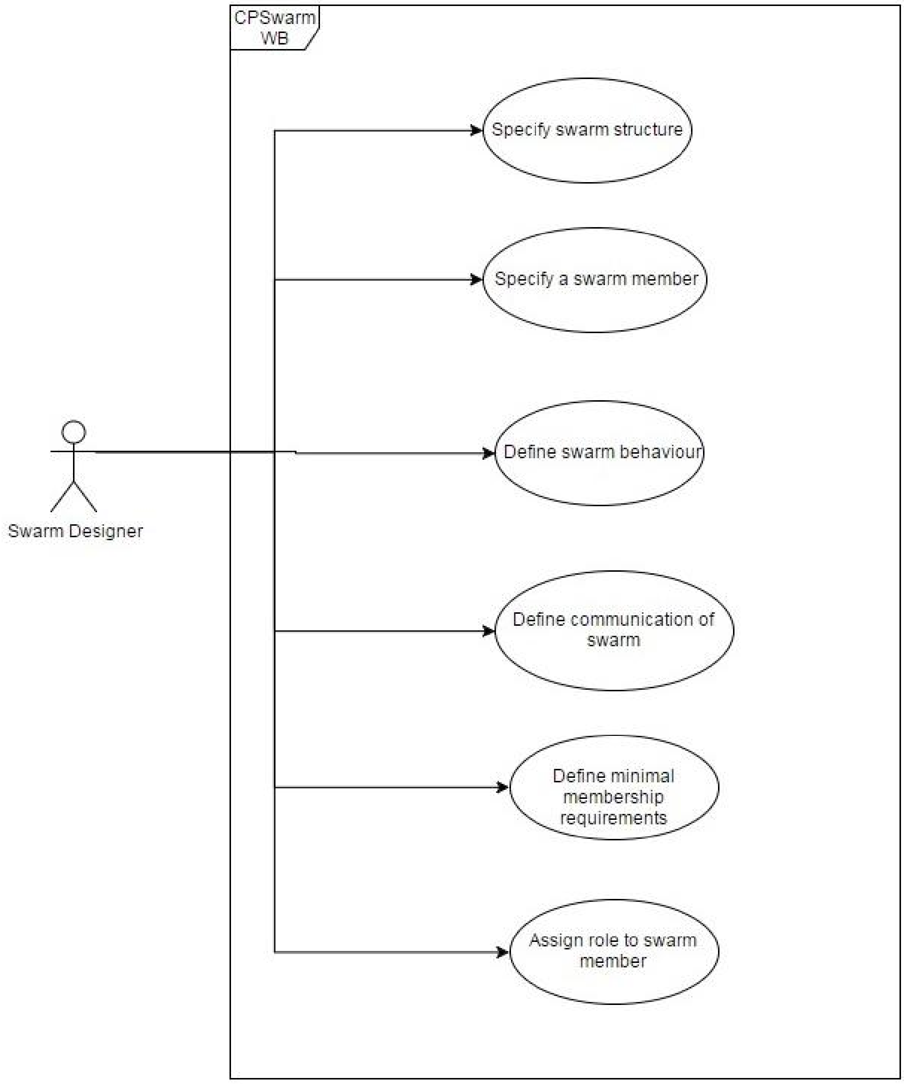


Figure 23: Swarm Designer use cases.

Use Case ID	UC-2.1
Use Case Name	Specify swarm structure
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 10
Involved Actors	Swarm Designer
Preconditions	A mission is defined and communicated to the involved actor(s).
Trigger	When a mission is communicated to the swarm designer.
Brief Description	This use case enables the involved actor(s) to specify the details related to the structure of the swarm according to the problem definition. Swarm structure includes the details of its components and the amount of components it consists of. The swarm designer analyses the mission details and decides which resources should be allocated in that particular situation.

Post-conditions	Swarm structure is successfully specified.

Use Case ID	UC-2.2
Use Case Name	Specify a swarm member
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 10
Involved Actors	Swarm Designer
Preconditions	• A problem has already been designed. • Swarm structure has already been specified.
Trigger	The need to add a member to a swarm
Brief Description	This use case enables the involved actor(s) to define the specifications of a swarm member. This swarm member may be a drone, rover or any other kind of a robot. In order to define a swarm member, the swarm designer has to specify the following: • Size (dimensions) • Sensors attached to the component • Behavior • Capabilities • Limitations This use case only specify the structural and behavioral details of a unit component irrespective of any external factors.
Post-conditions	A swarm member is successfully specified.

Use Case ID	UC-2.3
Use Case Name	Define swarm behaviour
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 10
Involved Actors	Swarm Designer
Preconditions	• Problem has already been defined • Swarm structure has been defined • Swarm members are already specified
Trigger	The need to define how a swarm would behave in different situations.
Brief Description	This use case refers to the ability of the involved actor(s) to define the behaviour of the swarm. The swarm designer defines both the internal and external behaviour of the swarm. Internal behaviour of a swarm refers to the details of how different members of the swarm would behave to coordinate with each other. Whereas, the external behaviour of swarm refers to how a swarm behaves

	collectively. The swarm designer also defines how a swarm would behave in case of a manual override of behaviour by the swarm operator.
Post-conditions	Swarm behaviour is successfully defined.

Use Case ID	UC-2.4
Use Case Name	Define communication of swarm
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 10
Involved Actors	Swarm Designer
Preconditions	• Problem has already been defined • Swarm structure has been defined • Swarm members are already specified • Swarm behaviour is already defined
Trigger	The need to define inter-communication between swarm members
Brief Description	This use case enables the involved actor(s) to define how various members of the swarm communicate with each other. This communication is hugely impacted by environment description provided by the mission planner and the domain limitations provided by the domain expert.
Post-conditions	Communication of swarm is successfully defined.

Use Case ID	UC-2.5
Use Case Name	Define minimal membership requirements
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 10
Involved Actors	Swarm Designer
Preconditions	• Problem has already been defined • Swarm structure has been defined • Swarm members are already specified • Swarm behaviour is already defined • Swarm communication is defined
Trigger	The need for an external component to join a swarm
Brief Description	This use case enables the swarm designer to specify the minimal requirements for an external component to be able to join a swarm. An example can be the automotive scenario (cf. Section 3.2) where a car desires to join a car convoy in order to become a swarm

	member until a specific destination is reached.
Post-conditions	The minimal requirements to be a member of a swarm are successfully defined.

Use Case ID	UC-2.6
Use Case Name	Assign role to swarm member
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 10
Involved Actors	Swarm Designer
Preconditions	• Problem has already been defined • Swarm structure has been defined • Swarm members are already specified • Swarm behaviour is already defined.
Trigger	The need to assign a role to a swarm member
Brief Description	This use case refers to the ability of the involved actor(s) to assign a role to a specific member of the swarm. For example, once a swarm structure is defined, the swarm designer realizes the need to declare one of the swarm members as the swarm leader. In this case, the swarm member that is declared the swarm leader will have some additional responsibilities.
Post-conditions	The desired role is successfully assigned to a swarm member.

5.3.4 Domain Expert

As described in Table 2, Domain Expert is a person, group or an organization who is an expert of the problem domain. He is responsible for providing expert advice about the domain e.g. rules, regulations, limitations etc. Figure 11 shows all the use cases related to the responsibilities of the Domain Expert, followed by their descriptions.

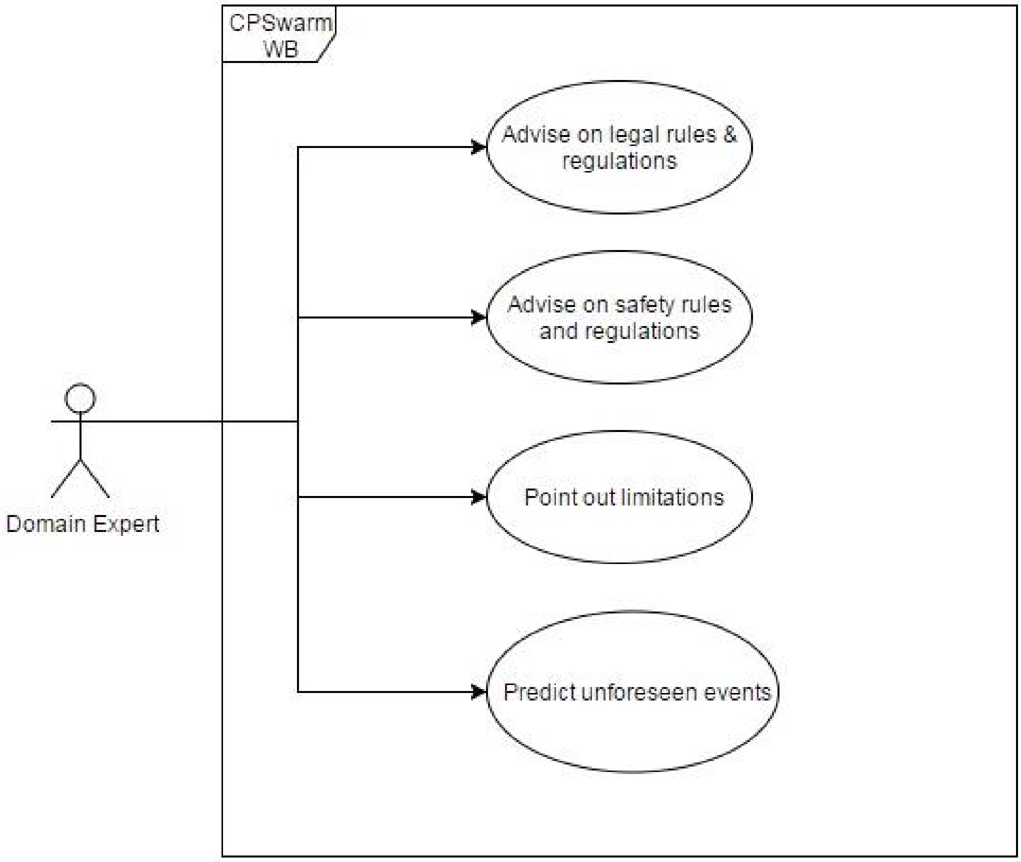


Figure 24: Domain Expert use cases.

Use Case ID	UC-3.1
Use Case Name	Advise on legal rules & regulations
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 11
Involved Actors	Domain Expert
Preconditions	Problem is already defined
Trigger	Need for advice on legal rules & regulations
Brief Description	This use case enables the involved actor(s) to provide advice on legal rules & regulations. The purpose of this use case is to make sure the Mission Planner is aware of the details of domain specific legalities while defining the mission.
Post-conditions	Advice on legal rules & regulations is successfully provided

Use Case ID	UC-3.2
Use Case Name	Advise on safety rules and regulations
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 11

Involved Actors	Domain Expert
Preconditions	Problem is already defined
Trigger	Need for advice on safety rules & regulations
Brief Description	This use case enables the involved actor(s) to provide advice on safety rules & regulations. The purpose of this use case is to make sure the Mission Planner is aware of the domain specific safety details while defining the mission.
Post-conditions	Advice on safety rules & regulations is successfully provided

Use Case ID	UC-3.3
Use Case Name	Point out limitations
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 11
Involved Actors	Domain Expert
Preconditions	Problem is already defined
Trigger	Need to point out limitations
Brief Description	This use case enables the involved actor(s) to point out domain specific limitations. It is important for the Mission Planner and Swarm Designer to know these limitations so that the planned mission and the swarm design created by these two adhere to these limitations.
Post-conditions	Limitations are successfully pointed out

Use Case ID	UC-3.4
Use Case Name	Predict unforeseen events
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 11
Involved Actors	Domain Expert
Preconditions	Problem is already defined
Trigger	Need to predict unforeseen events
Brief Description	This use case enables the involved actor(s) to predict unforeseen events. For instance, in case of a drone these unforeseen events may include flying birds, planes or even other attacking drones.
Post-conditions	Unforeseen events have successfully been predicted

5.3.5 Swarm Modeller

As described in Table 2, a Swarm Modeller is a person who constructs the model of the swarm. This model is the visual representation of the structure and behaviour of the swarm specified by the swarm designer. Figure 12 shows all the use cases related to the responsibilities of the Swarm Modeller, followed by their descriptions.

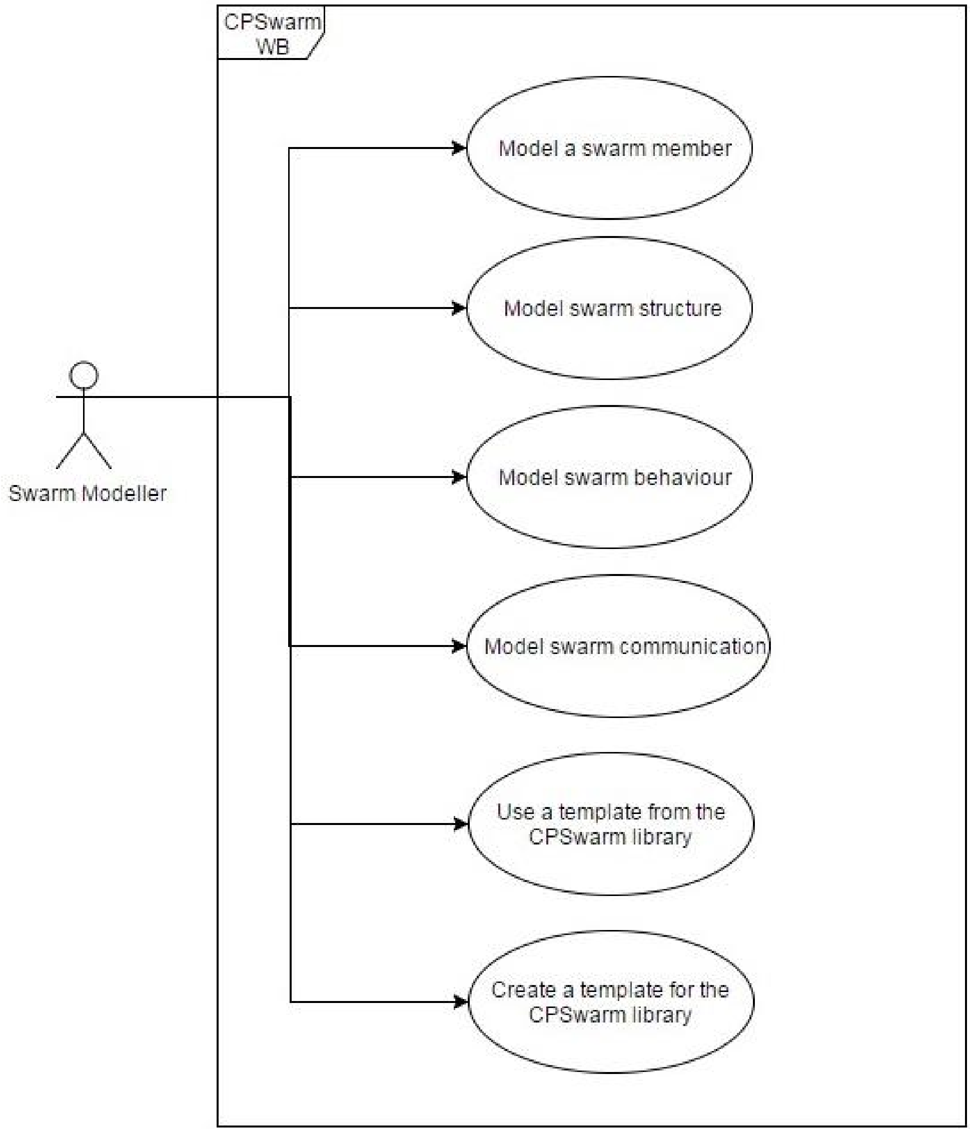


Figure 25: Swarm Modeller use cases.

Use Case ID	UC-4.1
Use Case Name	Model a swarm member
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 12
Involved Actors	Swarm Modeller
Preconditions	• Problem has already been defined • Swarm structure has been defined • Swarm members are already specified

	• Swarm behaviour is already defined
Trigger	Need to model a swarm member
Brief Description	This use case enables the involved actor(s) to model a swarm member. This swarm member may be a drone, rover or any other kind of a robot. This model of the swarm member is a visual model of the specifications provided by the swarm designer (UC - 2.2)
Post-conditions	A swarm member is successfully modelled

Use Case ID	UC-4.2
Use Case Name	Model swarm structure
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 12
Involved Actors	Swarm Modeller
Preconditions	• Problem has already been defined • Swarm structure has been defined • Swarm members are already specified • Swarm behaviour is already defined
Trigger	Need for modelling a swarm
Brief Description	This use case enables the involved actor(s) to model swarm structure. The swarm structure includes the details of various members of the swarm and number of each kind of member. This model is the visual model of the specification of swarm structure provided by the Swarm Designer (UC - 2.1).
Post-conditions	Swarm structure is successfully modelled

Use Case ID	UC-4.3
Use Case Name	Model swarm behaviour
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 12
Involved Actors	Swarm Modeller
Preconditions	• Problem has already been defined • Swarm structure has been defined • Swarm members are already specified • Swarm behaviour is already defined
Trigger	Need to model a swarm
Brief Description	This use case enables the involved actor(s) to model swarm behaviour. The swarm behaviour model includes the details of how the swarm would behave in case of normal/unforeseen circumstances and in case of manual override by the Swarm Operator. This model is the visual model of the specification of

	swarm behaviour provided by the Swarm Designer (UC - 2.3).
Post-conditions	Swarm behaviour is successfully modelled

Use Case ID	UC-4.4
Use Case Name	Model swarm communication
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 12
Involved Actors	Swarm Modeller
Preconditions	• Problem has already been defined • Swarm structure has been defined • Swarm members are already specified • Swarm behaviour is already defined
Trigger	Need to model swarm communication
Brief Description	This use case enables the involved actor(s) to model swarm communication. The swarm communication model includes the details of various members of the swarm would communicate with each other. This model is the visual model of the specification of swarm communication provided by the Swarm Designer (UC - 2.4).
Post-conditions	Swarm communication is successfully modelled

Use Case ID	UC-4.5
Use Case Name	Use a template from the CPSwarm library
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 12
Involved Actors	Swarm Modeller
Preconditions	• Problem has already been defined • Swarm structure has been defined • Swarm members are already specified • Swarm behaviour is already defined
Trigger	Need to use a template from CPSwarm library
Brief Description	This use case enables the involved actor(s) to use a template from the CPSwarm library. This template could be of a swarm member, behaviour, structure or communication. The purpose of this use case is to make sure that previously created models could be reused later.
Post-conditions	Desired template from the CPSwarm library is successfully used

Use Case ID	UC-4.6
Use Case Name	Create a template for the CPSwarm library
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 12
Involved Actors	Swarm Modeller
Preconditions	• Problem has already been defined • Swarm structure has been defined • Swarm members are already specified • Swarm behaviour is already defined
Trigger	Need to create a template for the CPSwarm library
Brief Description	This use case enables the involved actor(s) to create a template for the CPSwarm library. This template could be of a swarm member, behaviour, structure or communication. The purpose of this use case is to make sure that previously created models could be reused later.
Post-conditions	Template from the CPSwarm library is successfully created

5.3.6 Swarm Developer

As described in Table 2, a Swarm Developer is a person or a group responsible for adding logic to the generated code. This code is later on deployed on each component of the swarm. Figure 13 shows all the use cases related to the responsibilities of the Swarm Developer, followed by their descriptions.

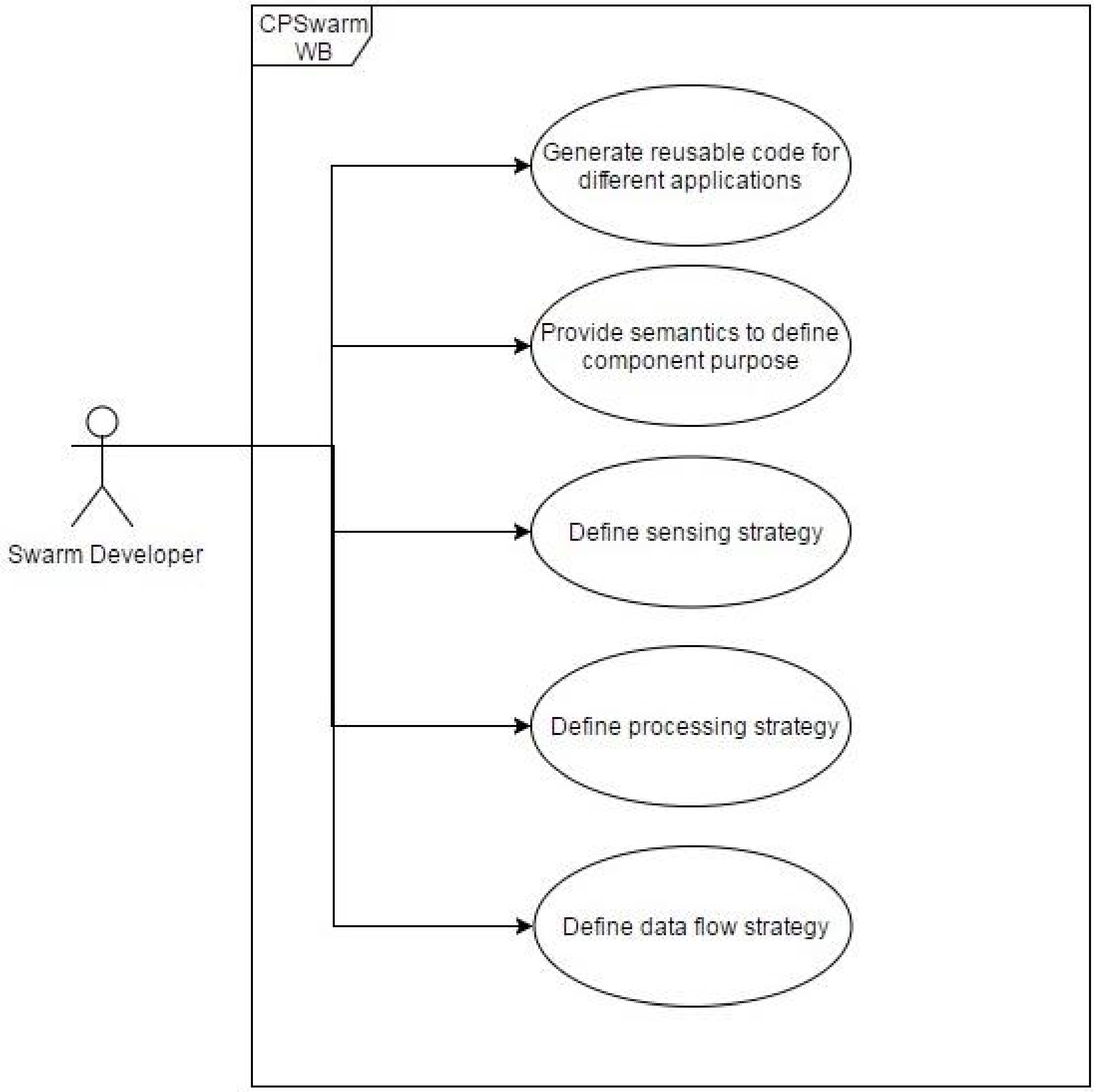


Figure 26: Swarm Developer use cases.

Use Case ID	UC-5.1
Use Case Name	Generate reusable code for different applications
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 13
Involved Actors	Swarm Developer
Preconditions	Swarm has already been modelled
Trigger	Need to generate reusable code for different applications
Brief Description	This use case enables the involved actor(s) to generate code that is reusable for different applications of the CPSwarm workbench.
Post-conditions	Reusable code for different applications is successfully generated

Use Case ID	UC-5.2
Use Case Name	Provide semantics to define component purpose
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 13
Involved Actors	Swarm Developer
Preconditions	Swarm has already been modelled
Trigger	Need to provide semantics to define component purpose
Brief Description	This use case enables the involved actor(s) to provide logic in code which defines the purpose of a particular component.
Post-conditions	Semantics to define component purpose are successfully provided

Use Case ID	UC-5.3
Use Case Name	Define sensing strategy
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 13
Involved Actors	Swarm Developer
Preconditions	Swarm has already been modelled
Trigger	Need to define sensing strategy
Brief Description	This use case enables the involved actor(s) to define a sensing strategy for swarm. The sensing strategy includes defining whether the sensing will be performed actively, passively or both. It also defines which sensors will be utilized for which purpose and for which kind of sensing.
Post-conditions	Sensing strategy is successfully defined

Use Case ID	UC-5.4
Use Case Name	Define processing strategy
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 13
Involved Actors	Swarm Developer
Preconditions	Swarm has already been modelled
Trigger	Need to define processing strategy
Brief Description	This use case enables the involved actor(s) to define a processing strategy for the swarm. For instance, processing strategy could be to use stream or edge.
Post-conditions	Processing strategy is successfully defined

Use Case ID	UC-5.5
Use Case Name	Define data flow strategy
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 13
Involved Actors	Swarm Developer
Preconditions	Swarm has already been modelled
Trigger	Need to define data flow strategy
Brief Description	This use case enables the involved actor(s) to define a data flow strategy for a swarm. Here data flow means the flow of data in the swarm i.e. where does the data go from the swarm.
Post-conditions	Data flow strategy is successfully defined

5.3.7 Algorithm Optimization & Simulation Expert

As described in Table 2, the Algorithm Optimization & Simulation Expert is a person or group who provides the expertise regarding the swarm algorithm. She/he decides the aptness of a certain algorithm given a specific swarm problem. Figure 14 and Figure 15 show all the use cases related to the responsibilities of the Algorithm Optimization & Simulation Expert, followed by their descriptions.

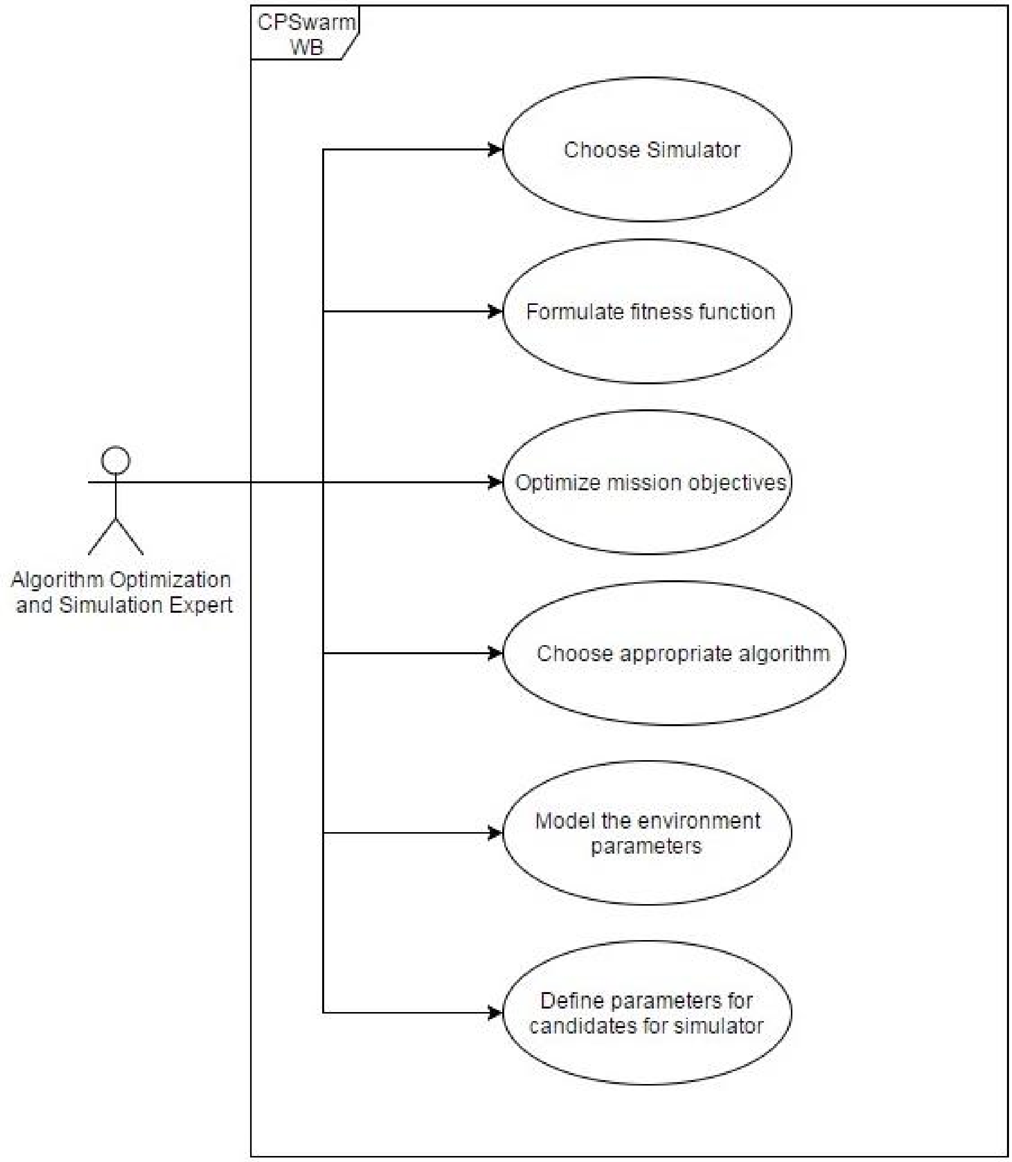


Figure 27: Algorithm Optimization & Simulation Expert use cases (1).

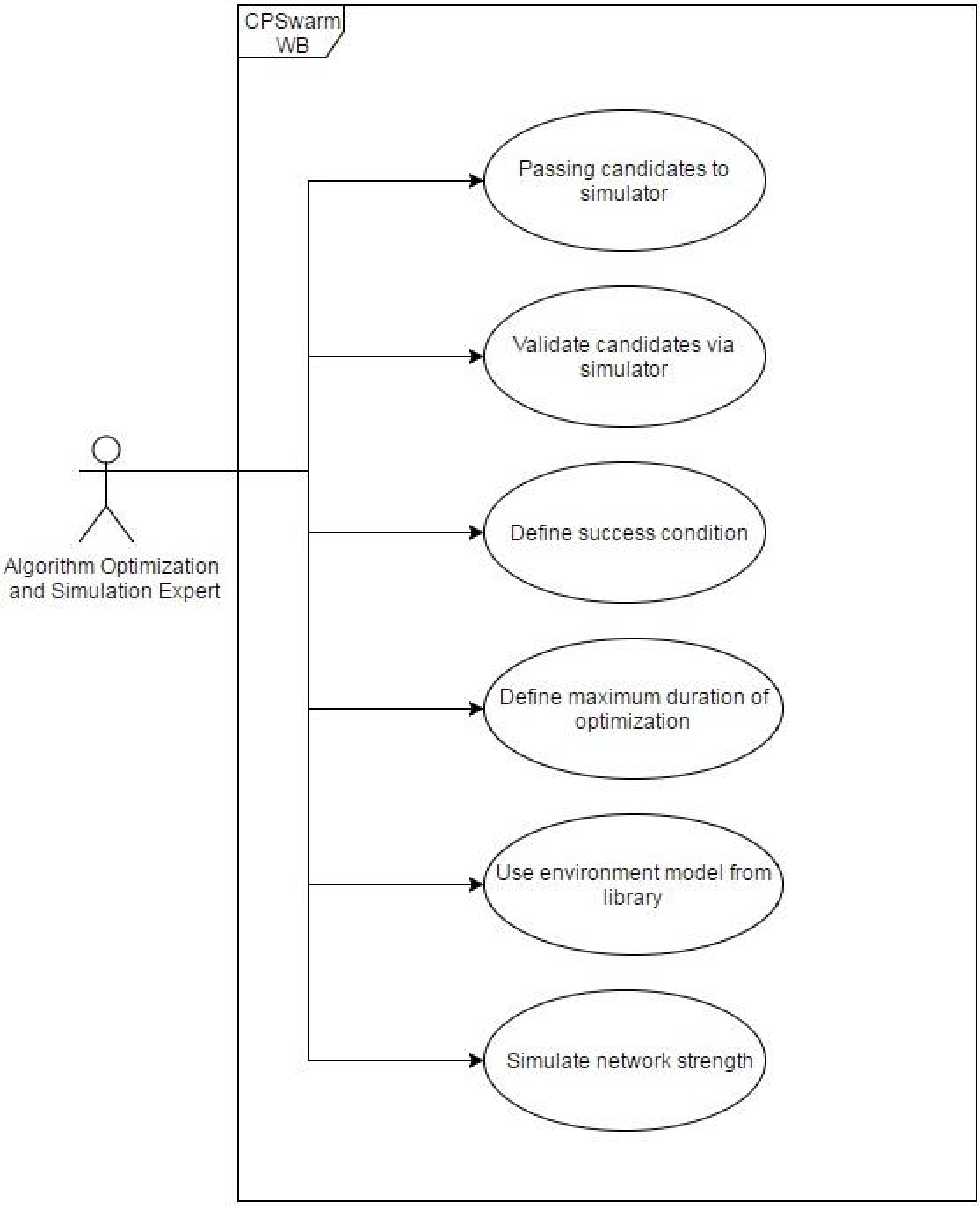


Figure 28: Algorithm Optimization & Simulation Expert use cases (2).

Use Case ID	UC-6.1
Use Case Name	Choose Simulator
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 14
Involved Actors	Algorithm Optimization & Simulation Expert
Preconditions	Swarm has already been modelled
Trigger	Need to choose a simulator
Brief Description	This use case enables the involved actor(s) to choose a simulator that will be appropriate for simulating a particular problem statement. The simulator will be used for algorithm optimization.
Post-conditions	Simulator is successfully chosen

Use Case ID	UC-6.2
Use Case Name	Formulate fitness function
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 14
Involved Actors	Algorithm Optimization & Simulation Expert
Preconditions	Swarm has already been modelled
Trigger	Need to formulate a fitness function
Brief Description	This use case enables the involved actor(s) to formulate a fitness function for algorithm optimization. This fitness function is a translation of the success condition of mission provided by the Mission Planner (UC - 1.5).
Post-conditions	Fitness function is successfully formulated

Use Case ID	UC-6.3
Use Case Name	Optimize mission objectives
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 14
Involved Actors	Algorithm Optimization & Simulation Expert
Preconditions	Swarm has already been modelled
Trigger	Need to optimize the mission objectives
Brief Description	This use case enables the involved actor(s) to optimize the success criteria of a mission, defined as objectives or by a fitness function for algorithmic optimization, if needed.
Post-conditions	Mission objectives are successfully optimized

Use Case ID	UC-6.4
Use Case Name	Choose appropriate algorithm
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 14
Involved Actors	Algorithm Optimization & Simulation Expert
Preconditions	Swarm has already been modelled
Trigger	Need to choose an appropriate algorithm
Brief Description	This use case enables the involved actor(s) to choose an algorithm that is apt for a given problem.
Post-conditions	An appropriate algorithm is successfully chosen

Use Case ID	UC-6.5
Use Case Name	Model the environment parameters
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 14
Involved Actors	Algorithm Optimization & Simulation Expert
Preconditions	Swarm has already been modelled
Trigger	Need to model environment parameters
Brief Description	This use case enables the involved actor(s) to model any parameters related to the environment. These parameters will be used for algorithm optimization. These environment parameters include the details of the environment description provided by the Mission Planner (UC - 1.3).
Post-conditions	Environment parameters are successfully modelled

Use Case ID	UC-6.6
Use Case Name	Define parameters for candidates for simulator
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 14
Involved Actors	Algorithm Optimization & Simulation Expert
Preconditions	Swarm has already been modelled
Trigger	Need to define parameters for candidates for simulator
Brief Description	This use case enables the involved actor(s) to define various parameters for each candidate to be passed on to the simulator. These candidates will be used for swarm algorithm optimization.
Post-conditions	Parameters for candidates for simulator are successfully defined

Use Case ID	UC-6.7
Use Case Name	Passing candidates to simulator
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 15
Involved Actors	Algorithm Optimization & Simulation Expert
Preconditions	Swarm has already been modelled
Trigger	Need to pass candidates to simulator
Brief Description	This use case enables the involved actor(s) to pass various candidates to the simulator. These candidates will be used for swarm algorithm optimization.
Post-conditions	Candidates are successfully passed on to the simulator

Use Case ID	UC-6.8
Use Case Name	Validate candidates via simulator
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 15
Involved Actors	Algorithm Optimization & Simulation Expert
Preconditions	• Swarm has already been modelled • A candidate is chosen as optimal
Trigger	Need to validate candidate via simulator
Brief Description	Once a candidate has been chosen as optimal, this use case enables the involved actor(s) to validate a candidate via the simulator.
Post-conditions	Candidate is successfully validated via simulator

Use Case ID	UC-6.9
Use Case Name	Define success condition
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 15
Involved Actors	Algorithm Optimization & Simulation Expert
Preconditions	Swarm has already been modelled
Trigger	Need to define success condition
Brief Description	This use case enables the involved actor(s) to define a success condition for swarm algorithm optimization. Once this success condition is reached, that particular candidate is considered as optimal.
Post-conditions	Success condition is successfully defined

Use Case ID	UC-6.10
Use Case Name	Define maximum duration of optimization
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 15
Involved Actors	Algorithm Optimization & Simulation Expert
Preconditions	Swarm has already been modelled
Trigger	Need to define maximum duration of optimization
Brief Description	This use case enables the involved actor(s) to define the maximum amount of time to be dedicated for the optimization of the swarm algorithm. This use case ensures that the optimization process does not continue once it has found the optimal candidate.
Post-conditions	Maximum duration of optimization is successfully defined

Use Case ID	UC-6.11
Use Case Name	Use environment model from library
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 15
Involved Actors	Algorithm Optimization & Simulation Expert
Preconditions	Swarm has already been modelled
Trigger	Need to use environment model from library
Brief Description	This use case enables the involved actor(s) to use a previously defined environment model from the respective library. The intention is to encourage reusability of environment models.
Post-conditions	Environment model is successfully used from library

Use Case ID	UC-6.12
Use Case Name	Simulate network strength
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 15
Involved Actors	Algorithm Optimization & Simulation Expert
Preconditions	Swarm has already been modelled
Trigger	Need to simulate network strength
Brief Description	This use case enables the involved actor(s) to simulate network strength if need be. This use case can be used for network stress testing purposes.
Post-conditions	Network strength is successfully simulated

5.3.8 Safety and Security Expert

As mentioned in Table 2 - Design-related stakeholders, the safety and security expert is responsible for the safety and security aspects of the swarm. She/he works closely with the system architect to specify the communication protocols to be followed by the swarm. He also ensures the detection of any security breach or malicious intrusions. In addition to these, he also formulates fall-back plans in case of emergencies. Figure 16 shows all the use cases related to the safety and security expert. Each use case is further described below.

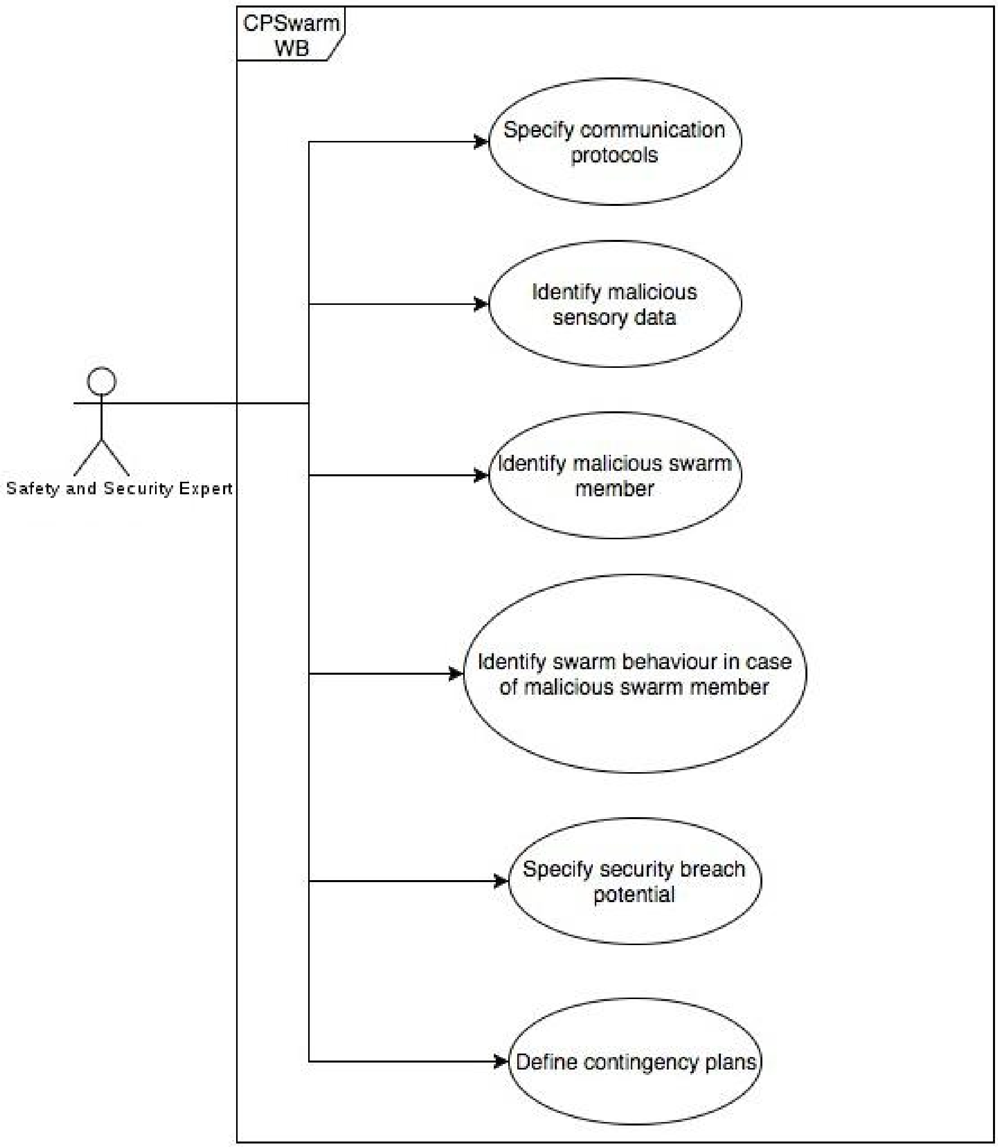


Figure 29: Safety and Security Expert use cases.

Use Case ID	UC-7.1
Use Case Name	Specify communication protocol
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 16
Involved Actors	Safety and Security Expert
Preconditions	• Problem has already been defined

	● Swarm structure has been defined ● Swarm members are already specified ● Swarm behaviour is already defined
Trigger	The need to specify protocols of communication between various swarm members
Brief Description	This use case enables the involved actor(s) to specify the protocols to be followed by the swarm members in order to communicate with each other. In case any swarm member fails to comply, it will be considered as a malicious member.
Post-conditions	Communication protocol is successfully specified.

Use Case ID	UC-7.2
Use Case Name	Identify malicious sensory data
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 16
Involved Actors	Safety and Security Expert
Preconditions	● Problem has already been defined ● Swarm structure has been defined ● Swarm members are already specified ● Swarm behaviour is already defined ● Swarm communication protocol is specified
Trigger	The need to identify if the data, acquired by various sensors of the swarm members, is corrupt or not.
Brief Description	This use case enables the involved actor(s) to monitor the integrity of the data acquired and transferred by all the swarm members. In case of malicious data, the safety and security expert is responsible for constructing a mechanism that is able to identify this malicious data to ensure security of the swarm communication.
Post-conditions	Malicious sensory data is successfully identified.

Use Case ID	UC-7.3
Use Case Name	Identify malicious swarm member
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 16
Involved Actors	Safety and Security Expert
Preconditions	● Problem has already been defined ● Swarm structure has been defined ● Swarm members are already specified ● Swarm behaviour is already defined ● Swarm communication protocol is specified

Trigger	The need to identify if any of the swarm members is corrupt or not.
Brief Description	This use case enables the involved actor(s) to identify if one of the swarm members has gone malicious. In order to ensure the safety and security of the swarm, it is extremely vital to quickly identify if a malicious swarm member exists.
Post-conditions	Malicious swarm member is successfully identified.

Use Case ID	UC-7.4
Use Case Name	Specify swarm behaviour in case of malicious swarm member
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 16
Involved Actors	Safety and Security Expert
Preconditions	• Problem has already been defined • Swarm structure has been defined • Swarm members are already specified • Swarm behaviour is already defined. • Swarm communication protocol is specified • Malicious swarm member is identified
Trigger	The need to specify swarm behaviour in case of malicious swarm member.
Brief Description	In case a malicious swarm member is identified, this use case enables the involved actor(s) to redefine the behaviour of the swarm according to the severity of the situation.
Post-conditions	Swarm behaviour in case of malicious swarm member is successfully specified.

Use Case ID	UC-7.5
Use Case Name	Specify security breach potential
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 16
Involved Actors	Safety and Security Expert
Preconditions	• Problem has already been defined • Swarm structure has been defined • Swarm members are already specified • Swarm behaviour is already defined • Swarm communication protocol is specified
Trigger	The need to avoid any security breach
Brief Description	This use case enables the involved actor(s) to thoroughly analyze the the swarm structure and communication protocol to identify

	how vulnerable it is to external breach. In order to construct a secure swarm, the potential for externals hacks and other security attacks should be next to none.
Post-conditions	The potential to encounter a security breach is successfully specified.

Use Case ID	UC-7.6
Use Case Name	Define contingency plans
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 16
Involved Actors	Safety and Security Expert
Preconditions	● Problem has already been defined ● Swarm structure has been defined ● Swarm members are already specified ● Swarm behaviour is already defined ● Swarm communication protocol is specified
Trigger	The need to define contingency plans
Brief Description	This use case enables the involved actor(s) to define fallback plans in case anything other than the ordinary happens. These events may include emergency situations, power or communication failure, external security breach or internal malicious behaviour. This use case ensures that in every foreseen or unforeseen situation, there exists a contingency plan.
Post-conditions	Contingency plans are successfully defined.

5.3.9 Deployer

As described in Table 2, Deployer is a person or group responsible for deploying the code of the swarm. Figure 17 shows all the use cases related to the responsibilities of the Deployer, followed by their descriptions.

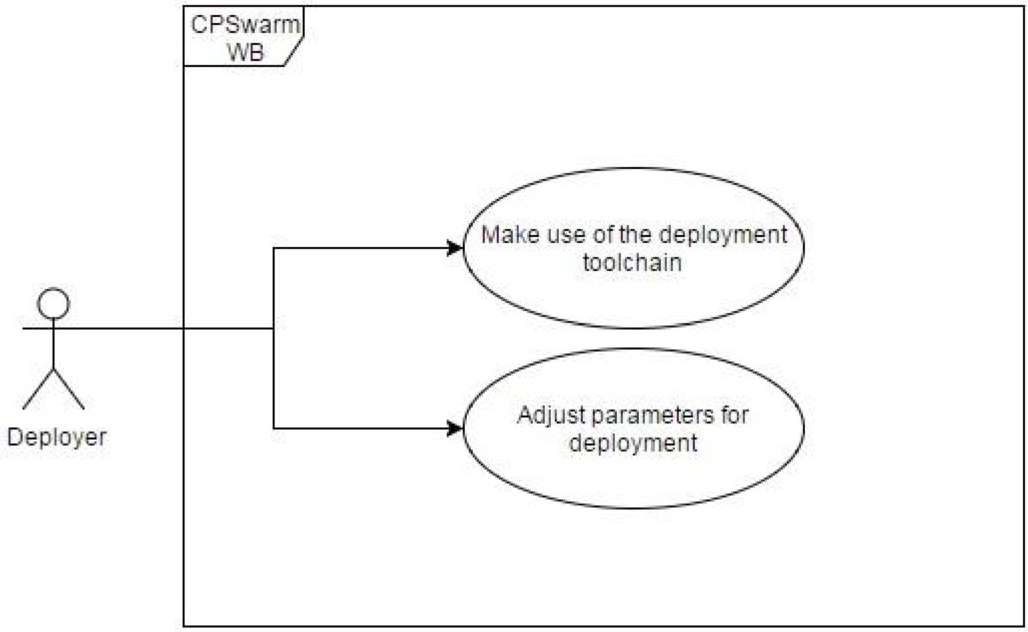


Figure 30 - Deployer use cases.

Use Case ID	UC-8.1
Use Case Name	Make use of the deployment toolchain
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 17
Involved Actors	Deployer
Preconditions	Swarm code has been developed
Trigger	Swarm code is ready to deploy
Brief Description	Once the code is developed and is ready to deploy, this use case enables the involved actor(s) to make use of the deployment toolchain.
Post-conditions	Deployment toolchain is successfully used

Use Case ID	UC-8.2
Use Case Name	Adjust parameters for deployment
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 17
Involved Actors	Deployer
Preconditions	• Swarm code has been developed • Swarm code is ready to deploy
Trigger	Need to adjust parameters for deployment
Brief Description	This use case enables the involved actor(s) to adjust the parameter for deployment according to the application.
Post-conditions	Deployment parameters are successfully adjusted

5.3.10 Swarm Operator

As described in Table 2, Swarm Operator is a person with the command control in his hand. He is responsible for directly manipulating the components of the swarm. Figure 18 shows all the use cases related to the responsibilities of the Swarm Operator, followed by their descriptions.

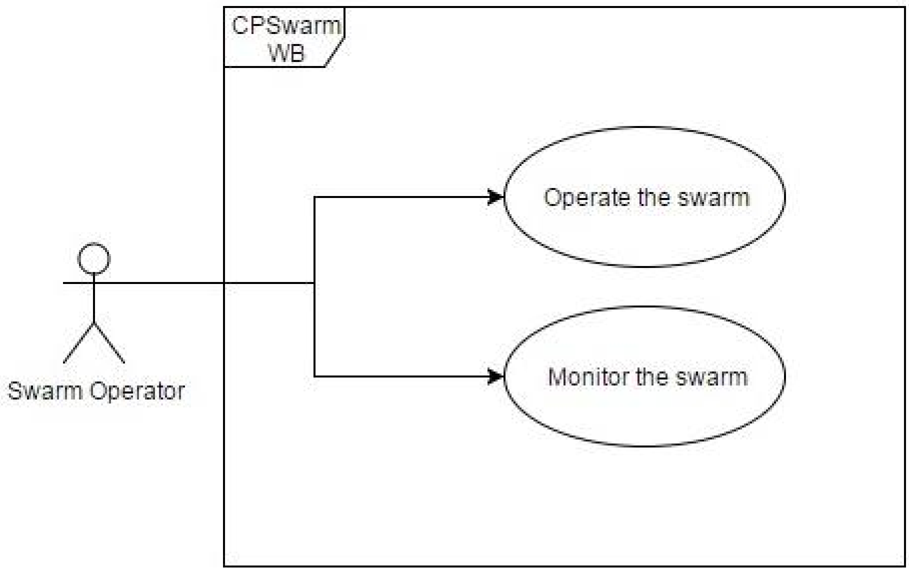


Figure 31 - Swarm Operator use cases.

Use Case ID	UC-9.1
Use Case Name	Operate the swarm
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 18
Involved Actors	Swarm Operator
Preconditions	• Swarm code has been deployed • Swarm is ready to be operated
Trigger	Need to operate the swarm
Brief Description	This use case enables the involved actor(s) to operate the swarm in real-time. The swarm may be controlled via remote control held by the Swarm Operator.
Post-conditions	Swarm is successfully operated

Use Case ID	UC-9.2
Use Case Name	Monitor the swarm
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 18
Involved Actors	Swarm Operator
Preconditions	• Swarm code has been deployed • Swarm is being operated
Trigger	Need to monitor the swarm
Brief Description	This use case enables the involved actor(s) to monitor the swarm in real-time. The swarm may be controlled via remote control held by the Swarm Operator.
Post-conditions	Swarm is successfully monitored

6 Conclusion

The initial phase of the CPSwarm project has focused on the specification of use cases, the definition of its stakeholders, as well as the description of the communication flow between them. Beyond, it addressed the workflow of the workbench and to illustrate how the deployment of CPSwarm workbench is envisioned in practice.

The objective of the present deliverable was to establish a common ground on which the remaining WP2 tasks, and later the remaining technical WPs (WP3 to WP7), will build their foundations towards the demonstration (WP8). The work in WP2 follows a scenario-driven approach, starting with the formulation of vision towards which the project will develop. The visions serve as basis for identifying involved stakeholders, available knowledge, used technologies as well as their interplay and data flow. From the basic set of use cases, further specifications of workflows performed with the help of the CPSwarm workbench will evolve. The use cases specified in this version of the deliverable are to be seen as a first iteration and will be revised and further refined in the following version of the deliverable, in the scope of remaining WP2 tasks and WPs 3 to 8. By defining a common set of use cases, this deliverable D2.1 laid the foundation for creating the initial set of requirements in D2.3 that will be used in further implementation in the technical WPs.

The analysis presented in this deliverable started with the description of the extensive brainstorming session. The analysis of the information attained from this brainstorming session resulted in the generation of three vision scenarios, detailed domain analysis, identification of purpose and workflow of the workbench, initial set of stakeholders, categorization of these stakeholders, the communication flow between them and lastly, a comprehensive set of use cases grouped with respect to the involved actors.

Acronyms

Acronym	Explanation
UC	Use Case
UAV	Unmanned Arial Vehicle
SAR	Search and Rescue
ROS	Robot Operating System

List of figures

List of tables

References

[1] K. van der Heijden, Scenarios: The Art of Strategic Conversation, Wiley, Chichester, 1996.
[2]Pixhawk website: https://pixhawk.org/modules/px4io
[3] PX4 Development Guide (Wiki): https://pixhawk.org/modules/px4io
[4] The ROS Framework : http://www.ros.org [last visit date 15 March 2017]
[5] Gazebo : http://gazebosim.org [last visit date 15 March 2017]
[6] RAWFIE : http://www.rawfie.eu [last visit date 15 March 2017]
[7] ROS Components : https://www.roscomponents.com/en [last visit date 15 March 2017]
[8] GitHub Repository (RobotnikAutomation) : https://github.com/robotnikAutomation [last visit date 15 March 2017]
[9] Lewis, C., Polson, P. G., Wharton, C., & Rieman, J. (1990). Testing a Walkthrough Methodology for Theory-Based Design of Walk-Up-and-Use Interfaces. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 235-242). New York, NY, USA: ACM. doi:10.1145/97243.97279
[10] Cockburn, A. (2000). Writing Effective Use Cases. Addison-Wesley Professional; 1st edition
[11] ISO 26262. Road vehicles – Functional safety. International standard for functional safety of electrical and/or electronic systems in production automobiles defined by the International Organization for Standardization (ISO) in 2011.
[12] Rviz. 3D visualizer for ROS. (http://wiki.ros.org/rviz)
[13] Modelio. An open source UML / BPMN modeling tool. (https://www.modelio.org)