Ref. Ares(2017)2880528 - 08/06/2017

D2.1 – INITIAL VISION SCENARIOS AND USE CASE DEFINITION

Deliverable ID	**D2.1**
Deliverable Title	**Initial Vision Scenarios and Use Case Definition**
Work Package	**WP2**
Dissemination Level	**PUBLIC**
Version	**2.0**
Date	**2017-05-31**
Status	**Final**
Lead Editor	**Sarah Suleri (FRAUNHOFER)**
Main Contributors	**Sarah Suleri, René Reiners (FRAUNHOFER), Edin Arnautovic (TTTECH), Omar Morando (DGSKY), Etienne Brosse (SOFTEAM), Melanie Schranz (LAKE), Rafa Lopez (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	2017-04-10	Sarah Suleri (FRAUNHOFER)	First Draft with relevant materials
0.2	2017-05-10	Sarah Suleri , René Reiners (FRAUNHOFER)	Revised draft / reworked structure + structured brainstorming analysis
0.3	2017-05-12	Sarah Suleri , René Reiners (FRAUNHOFER)	Completed Use Case Diagrams and Descriptions
1.0	2017-05-15	Sarah Suleri , René Reiners, (FRAUNHOFER), Omar Morando (DGSKY), Edin Arnautovic (TTTECH)	Proof-Read version ready for review
2.0	2017-05-31	Sarah Suleri, René Reiners (FRAUNHOFER)	Reworked version ready for submission to EC

Internal Review History

Review Date	Reviewer	Summary of Comments
2017-05-30	Omar Morando (DGSKY)	Approved: • Minor issues to fix
2017-05-30	Edin Arnautovic (TTTECH)	Approved: • Minor issues to fix

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.

Deliverable D2.1 is compiled with a collaborative effort of all partners who actively participated in technology demonstrations conducted for domain analysis and brainstorming sessions conducted for the stakeholder and use case analyses. This document reports on the iterative process of ideation which resulted in the identification of the purpose and workflow of the workbench, an 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.

The use cases specified in this deliverable should be seen as a first vision towards which the project will evolve. They are expected to be revised, extended and refined in the future iteration of the deliverable that accompanies the project.
Currently, the visions are used for inter-project communication in order to derive development scenarios and help to fertilize the process of thinking in future systems. 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 initial requirement analysis to be documented in D2.3 in WP2, and later for the remaining technical WPs (WP3 to WP7), towards the demonstration (WP8).

Table of 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 create and refine vision scenarios that will be used during the elicitation of user requirements for the different use cases and stakeholders involved. The use case specification will be provided for:

- Swarm of drones
- Swarm Logistics
- Autonomous freight vehicles

This document describes the activities to support the identified workbench workflow, adapting it to the different environments involved in the CPSwarm project and provide a thorough analysis of the candidate 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 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 will be continuously updated and refined through an iterative process that will lead to the production of two releases of this document, which the second one planned for project month M16. The development of this deliverable was coordinated by FRAUNHOFER with contribution of DiGiSky, TTTech, Robotnik, KLU, ISMB, SOFTEAM, LAKE and SLab. The outcome of this deliverable will be used for deliverable D2.3: *Initial Requirements Report,* due in M6. Discussions held with the partners on currently available technologies and applied processes belong to efforts in both initial work packages WP2 and WP3. Thus, the results are partly reported in this deliverable and serve as starting point for describing available components and their interplay in D3.1.

1.1 Related documents

ID	Title	Reference	Version	Date
D2.3	Initial Requirements Report			M6
D3.1	System Architecture Analysis & Design Specification			M6
D2.1v2	Updated Vision Scenarios and Use Case Definition			M16

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.

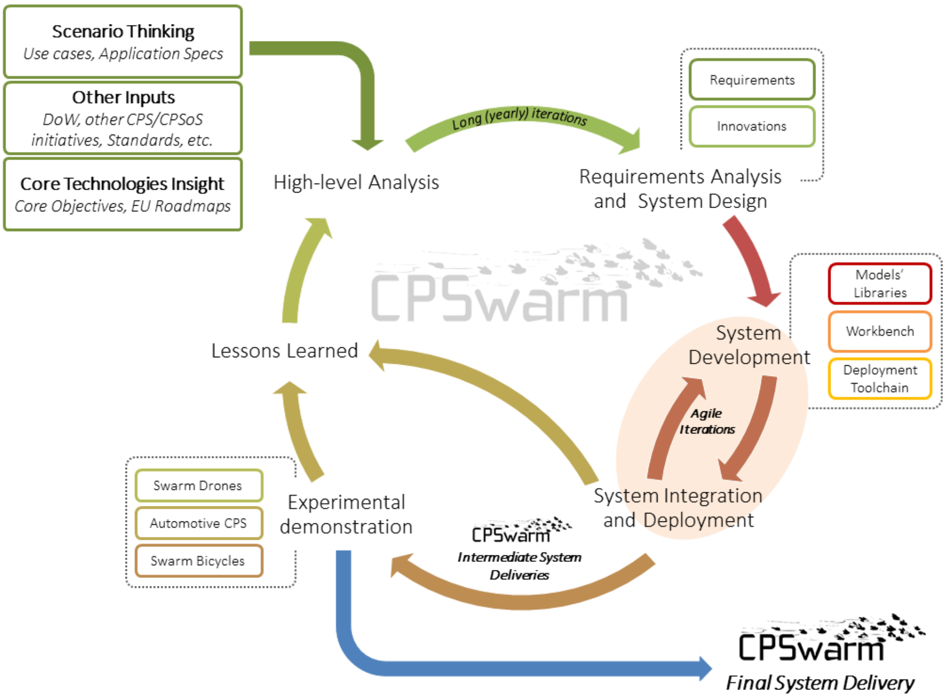

Figure 1: The CPSwarm Workbench development lifecycle.

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, *vision scenarios* were created for the different domains targeted in the CPSwarm project: (i) Surveillance and Emergency Response, (ii) Automotive and (iii) Logistics. 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 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 Vision Scenarios

Scenarios are a free form narrative that tells the existing practices as well as future visions of a product or system being developed [1]. Scenarios normally do not explain the details of the interaction or the system; they are purposefully left rough and vague in order to avoid users hanging onto details and overlooking the general issue of whether the vision expressed by the scenario fits in their world and fulfils their needs.

The following sections describe the three vision scenarios. These vision scenarios briefly explain the three different domains and how things would work if the CPSwarm workbench were used. An initial version of the scenarios was provided after internal discussion at proposal submission. During the kick-off-event in Turin, the scenarios were discussed again and refined. The current versions as described in the following sections will be used in the first iteration of the CPSwarm development cycle (cf. Figure 1).

3.1 Swarm Drones

In CPSwarm, we will consider heterogeneous swarms of ground robots/ rovers and UAVs to conduct certain missions in the surveillance of critical infrastructure e.g., industrial or power plants as well as in Search and Rescue (SAR) tasks.

The Swarm Drones scenario covers the following SAR mission:
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 analyses that there is a need to send in a swarm of 15 drones and 8 rovers to provide full coverage. The swarm goes into the fire, 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.

On top of this, members of human response teams may access the swarm and its data directly via wearable devices and, possibly influence/modify the swarm tasks. A possible test setting may be a mixed swarm of 4 UAVs and 4 rovers. They may be sent out to map a (disaster) area, look for casualties or perform intrusion detection with automated tracking of the intruder.

Thus, for surveillance, we envision applications of swarms for:

- intrusion detection, i.e. detection of unauthorized persons entering the plant area or
- follow and observe actions of unauthorized persons in the plant.

Apart from that in SAR tasks swarms can be exploited for:

- generating a situation overview of the disaster scene in case of an industrial plant accident including real-time images (VIS, IR), toxic and explosive gas leakage detection or
- finding human casualties or persons trapped in the disaster area.

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 Automotive CPS

The aim of the automotive use case in CPSwarm is to set up a laboratory level demonstrator (TRL 3 to TRL 4, demonstration in breadboard lab environment) in the area of autonomously driving vehicles intended for freight vehicles like trucks, vans or cars and connecting them via kind of an electronic *drawbar*. The focus of the research will be the technical implementation of such drawbar over wireless connection. For the vehicles to react appropriately to the information from other vehicles, the information must be reliable and timely predictable (deterministic real-time). Following scenario could be simulated: The leading vehicle prescribes the actions and decisions (i.e. navigation, decision on take-over manoeuvres, sequencing manoeuvres, lane change etc.) for the follow-up vehicle(s) that will make use of the leading vehicle´s actions. The follow-up vehicle would need full autonomous driving capability and environmental awareness as well. Environmental awareness will be simulated using e.g., pre-recorded video data for sensor input and sensor-fusion functions for object recognition.

TTTECH already provides computers for advanced driving assistance systems (ADAS) in a centralized computing architecture for European passenger car manufacturers. Next generations of such systems will be developed and integrated for the demonstration.

For the Automotive CPS, the following scenario is used as starting point:
Christian lives in Cologne. He wants to visit his sister in Munich over the weekend. Instead of driving alone, he decides to join the swarm from Cologne to Munich. The swarm details show that the swarm of cars will reach Cologne Central Station at 9 o'clock on Saturday morning. He thinks that the timing is perfect because he will reach Munich just in time for delicious homemade lunch. He packs his bags on Friday night and leaves his home early Saturday morning to reach the swarm in time. He reaches the central station 5 minutes early and decides to wait in his car. The swarm reaches the station at 9 and Christian joins the swarm to start his journey towards Munich. The directions and speed control is handled by the alpha car of the swarm. Christian is really enjoying his stress-free journey to Munich. A few miles before Wurzburg, the alpha car is notified that there is constructional work going on the road from Wurzburg to Nuremburg. The alpha car decides to take an alternative route through Bamberg. Since the swarm has grown during the trip, an intelligent routing mechanism recommends three alternate routes, each meant for a part of the swarm in order to avoid congestions in smaller streets. Thus, the swarm splits up and two new alpha cars are designated. Each of the three parts follows its alpha car. After the detour, dynamic speed management is used in order to merge the three parts of the original swarm, which then continues the journey to Munich. Christian is very happy that he does not have to worry about the route details but on demand, he is informed by the system why the detour and split up was performed. The swarm reaches Munich in 5 hours and 33 minutes. Christian decides to leave the swarm at Munich Central Station. Christian notifies the alpha car and leaves the swarm to go to his sister's house.

3.3 SWARM Logistics Assistant

The SWARM Logistics Assistant use case 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 3: 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.

For the Warehouse and Logistics domain, the following scenario is considered:

Alex has been working in a warehouse for the last 5 years. His main job is to shift a package from one point to another. Every working day, his day began by getting a list of packages he needed to shift. Sometimes these packages were very large and it was difficult for Alex to pick them up all by himself. He felt he needed help in such situations. His prayers were about to be answered. The warehouse decided to install CPSwarm robot assistance for moving the packages.

After the installation, things have become quite simple for Alex. Whenever he encounters a scenario where the package to be picked up is too big and/or heavy for him, he calls for assistance. Today was one of those days. Alex was assigned to pick up a very large package. He knew he needed help. He called an assistive robot for help. The assistive robot upon receiving Alex's call for help, located him in the warehouse using his knowledge of 3D maps of the warehouse.

Once the assistive robot reached Alex, he scanned the package and assessed how much help he needed to pick up that package. The robot analysed that the package was big enough to be carried by 3 more robots. He called 3 more for help. Once all robots reached Alex, they picked up the package and put it at the destination.

Alex is very happy with his job now because he knows that help is just one call away.

4 Technology Exploration

The technology exploration served as foundation for the ideation phase described in Chapter 0 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 Pix Hawk.
- 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.
- 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.

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

- The maximum distance fro video transmissions is up to 1.7km.

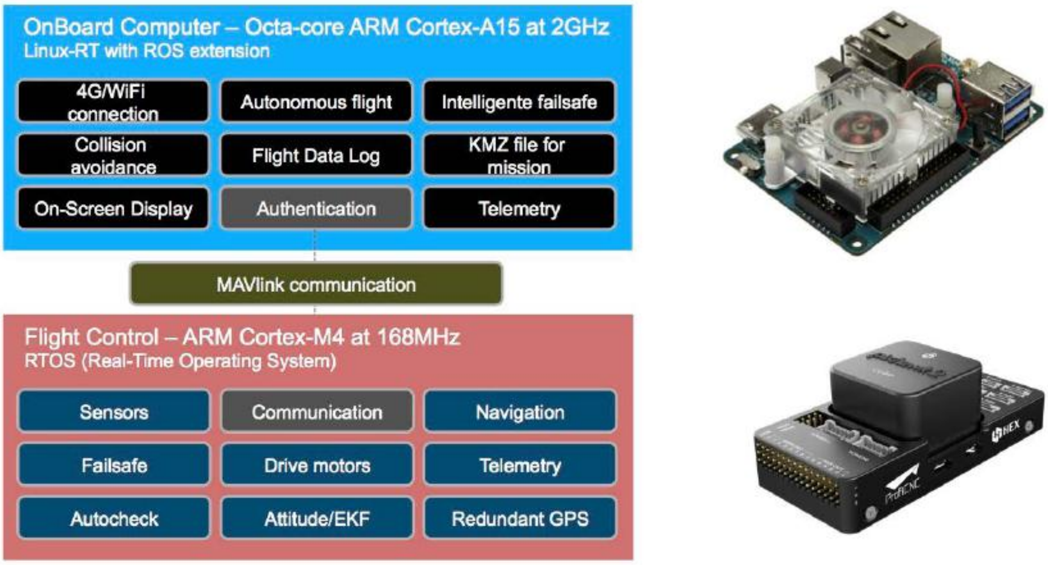

Figure 4 – DigiSky's UAV architecture.

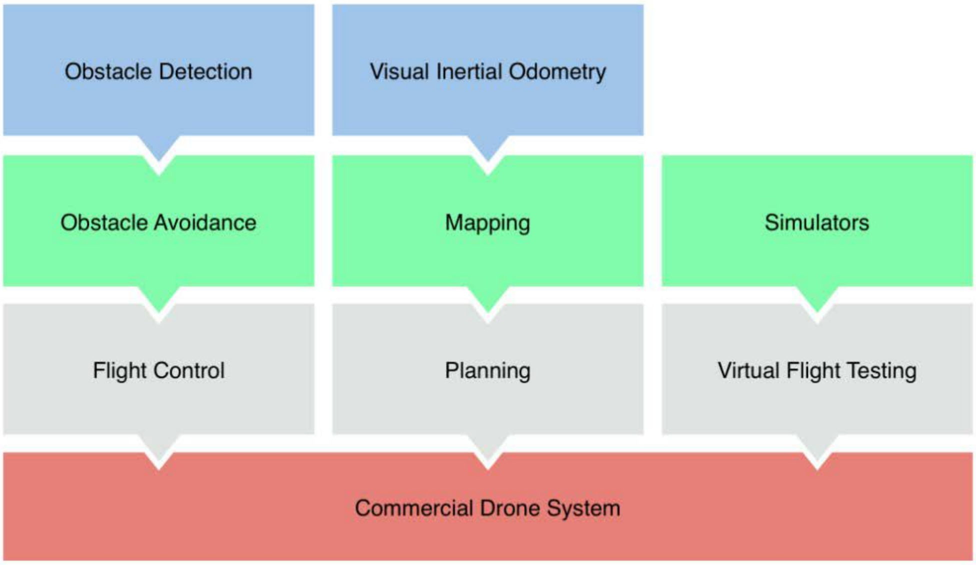

Figure 5 – Overview of flight control and auto-piloting features.

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 (each tools runs independently):

- Cmake
- Android tools

- 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.
- ROS nodes are represented as executable files
- Deployment is currently performed on ground via USB stick (rotors need to be removed --> safety).
- 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

During the technology demonstration by TTTech, the architectural overview of the laboratory setup of an automotive system was given. This setup (cf. Figure 6).) consists of a central computer, (1) an *automotive fog-node,* which is virtualized high-performance computer, envisioned for the future car architectures, (2) automotive electronic control units (ECUs), and (3) an deterministic Ethernet switch.

4.2.1 Used Tools and Applied Processes

The idea of the fog node is to bring some computational power and data processing closer in the vehicle and "offer" this processing power in a real-time manner to other units in the car, 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). Trays are available for extending the fog node with needed hardware.

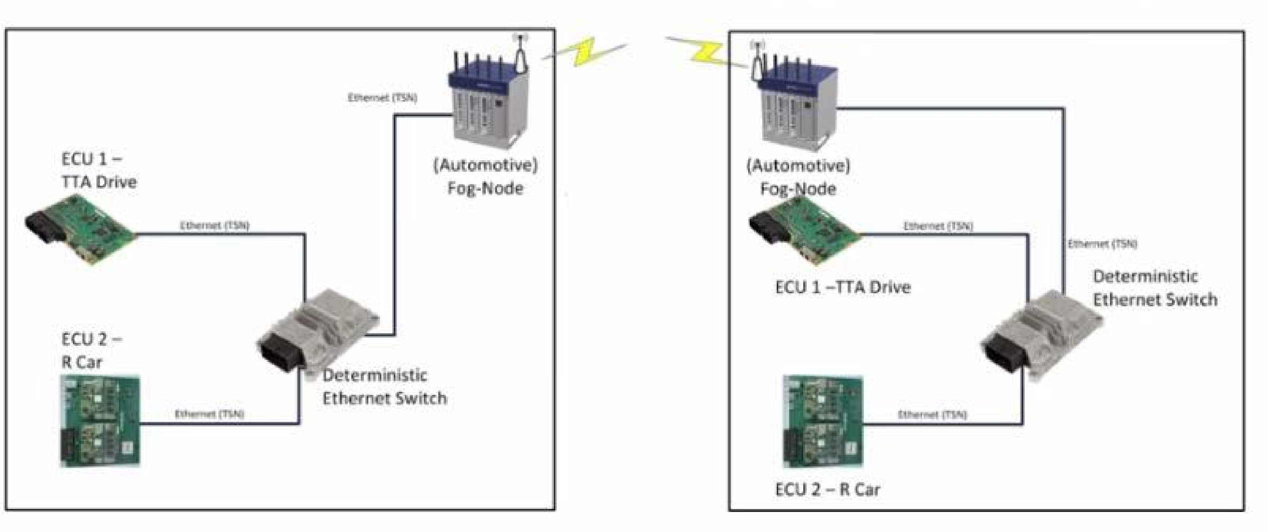

Figure 6: TTTech's Fog node & ECU.

- The fog node operates on standard Intel x86 CPU and standard hard drive and RAM.
- The fog node utilizes hypervisor (KVM), so that multiple applications can run in their own VM environments.
- Cores of the fog node can be dedicated to specific function --> guarantee of processing / prioritization.
- We can treat the fog node as an ordinary PC (with some customizable extensions for real time functionality) for deployment sake, which means deployment using Docker or other ordinary way could work.

There were also some discussions about real time scheduling, and the answer is that the fog node with hypervisor is capable of running real-time operating systems and is quite flexible in dealing with different real time requirements. It is also mentioned that communication between VMs is also possible.

Applications deployed on the automotive demonstrator are typically distributed. Such part of the software can be deployed on the Fog-Node and part on the ECUs. For example, pre-processing of sensor data, which requires fast real-time response but not much computing power can be deployed on the ECUs and advanced object recognition algorithms on the Fog-node. ECUs utilize AUTOSAR environment, where the communication between single components occurs over TTTech middleware (TTIntegration), which sits on top of the deterministic Ethernet on-board network. Software development in this way is much more complicated than the deploying on fog node:

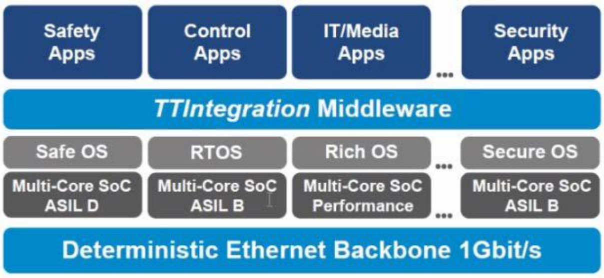

Figure 7 – TTTech view on a swam member's architecture.

- Firstly, if a customer wants to develop his own software component to be integrated into this platform, they will have to do some software modelling according to AUTOSAR specification (with the help of tools e.g. DaVinci Developer), which should specify the platform to be integrated in and the data types to communicate etc.
- Then TTTech will have to go through 29 steps semi-automated steps internally to integrate the software component into the platform.
- Lastly is the final integration into the platform. The process takes around two months to finish.

TTIntegration is utilized for the time-triggered real-world integration of software components from multiple vendors, where the fulfilment of hard real-time and safety requirements is critical. For the demonstration of CPSwarm concepts however, deployment without TTIntegration is more feasible, as there are many requirements for the software. TTTech also offered opportunity for training in how to do the software modelling.

4.2.2 Handover - Points

Replying to a question regarding their expectation about what to get for the code logic, TTTech said they could use C code, or black-box library, which provides C interface to map to the middleware.
In addition, firmware or software updates will be performed by connecting the ECU to the PC. For security issues, updates over the air should be avoided and only be possible in combination with a certified vendor.

4.2.3 Limitations

- TTTech mentioned that they want to automate the existing 29 steps needed for software integration.
- We raised the question, that if they want to replace the modelling process with Modelio, TTTech answered that it would be good to have this functionality, but it could be challenging.
- Another question was raised that maybe we should just focus on deploying on fog node, and TTTech answered that while it would be a conservative option to go for in the beginning, it would also be nice, if we manage to get it to be deployed on HADP.
- The currently used hardware is currently only available as testbed needing 220V current.
- The energy consumption is still quite high although smaller versions of the fog node are planned that are based on Atom technologies.
- The fog node can be used stationary --> future plans for turning it into a mobile version

4.2.4 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 (ROBOTNIK, ISMB: Please check this)
- 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 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.

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.

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 many 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.

[1] URDF : **http://wiki.ros.org/urdf**
[2] **http://www.robotnik.eu/mobile-robots/turtlebot-ros/**

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 [0] was also mentioned as an example of connecting robots from Robotnik with other vehicles, e.g. drones.

4.3.4 Further Reading

- The **ROS Framework** [4]
- **Gazebo5** [5]
- AGVS
- The **RAWFIE Project** [6]
- **ROS Components** [7]
- **GitHub Repository (Robotnik Automation)** [8]

5 Ideation for the CPSwarm Workbench Usage

Ideation is a creative process of having discussions and conducting brainstorming session in order to generate new ideas and attain a deeper understanding about a given scenario. After gaining insights on the available technologies and actions performed for realizing a swarm scenario, a CPSwarm Workbench Usage Scenario was derived. Therefore, workshops and brainstorming sessions were held during the consortium meeting in March at Fraunhofer premises.
This way, it became possible to document actions performed by different roles, data formats and information used, gaps between handing over to the next part as well as dependencies between different workbench components that need to be taken into account for defining harmonious flows for the workbench interaction. The latter had implications on the CPSwarm Workbench Architecture (see Deliverable D3.1).

In the following section, the topics and results from the brainstorming sessions are briefly described. The structured and summarized results are presented as use cases in Section 0.

5.1 Brainstorming

The starting point for the brainstorming session, a "CPSwarm Workbench Scenario" was given. This time, instead of envisioning what becomes possible if the CPSwarm Workbench was available (project goal) the scenario treated the question on how a development team would work with the workbench along a given vision scenario. In group work, different development steps starting from scenario analysis, gathering of context information over modelling, development, simulation and deployment were followed on a cognitive walkthrough-basis [9].

As underlying scenario, the realization of a drone surveillance scenario was given as tasks. The scenario is similar to the visions scenario described in Section 3.1. However, the aspect of changing the swarm's behaviour for handling emergencies was left out for simplification in this stage of the project.
Each workshop member was now asked which steps he would undertake in order to realize the scenario with his available technologies. This way, all participants were shown the available technologies applied to a realistic scenario. When one step in the development process (represented by different partners) ended, the handover to a, still visionary, new component of the CPSwarm workbench was simulated.

The following questions stimulated the brainstorming session:

- What are existing technologies and workflows you make use of when realizing the scenario?
- When is your work over, i.e., when would you need to hand over to another technology?
- Where are limitations? What would you need next such that your work is improved or facilitated?
- What components of the workbench would you use?
- For what purpose and in which context would you use the workbench?
- What is a meaningful workflow of the workbench? (cf. Section 6.1)
- What different components of the workbench do you identify?
- What are data and information handover points of the workbench components?
- What roles are involved in the development process and what are their responsibilities? (cf. Section 6.2)
- How would you interact with the workbench?

In the following section, the results of the brainstorming session will be briefly summarized. All results were taken into account for processing afterwards by the requirements engineering team in order to formulate, stakeholder, roles, actors and use cases in the scope of the use case analysis (cf. Section 0).

5.2 Swarm Engineering

Figure 8 shows the gathered input from the brainstorming session at a glance alongside identified steps for modelling, simulating, optimizing and deploying a swarm application.

Figure 8: Brainstorming results for different engineering steps.

5.3 Modelling

From the feedback on these steps, further topics were derived. Figure 9 and Figure 10 show the notes gathered around the topic of modelling that reach further beyond the model of swarm members. This way, the swarm structure, its behaviour and communication model needs to be taken into account. For simulations and optimizations, the modelling of the environment and its adjustment when prototyping becomes important.

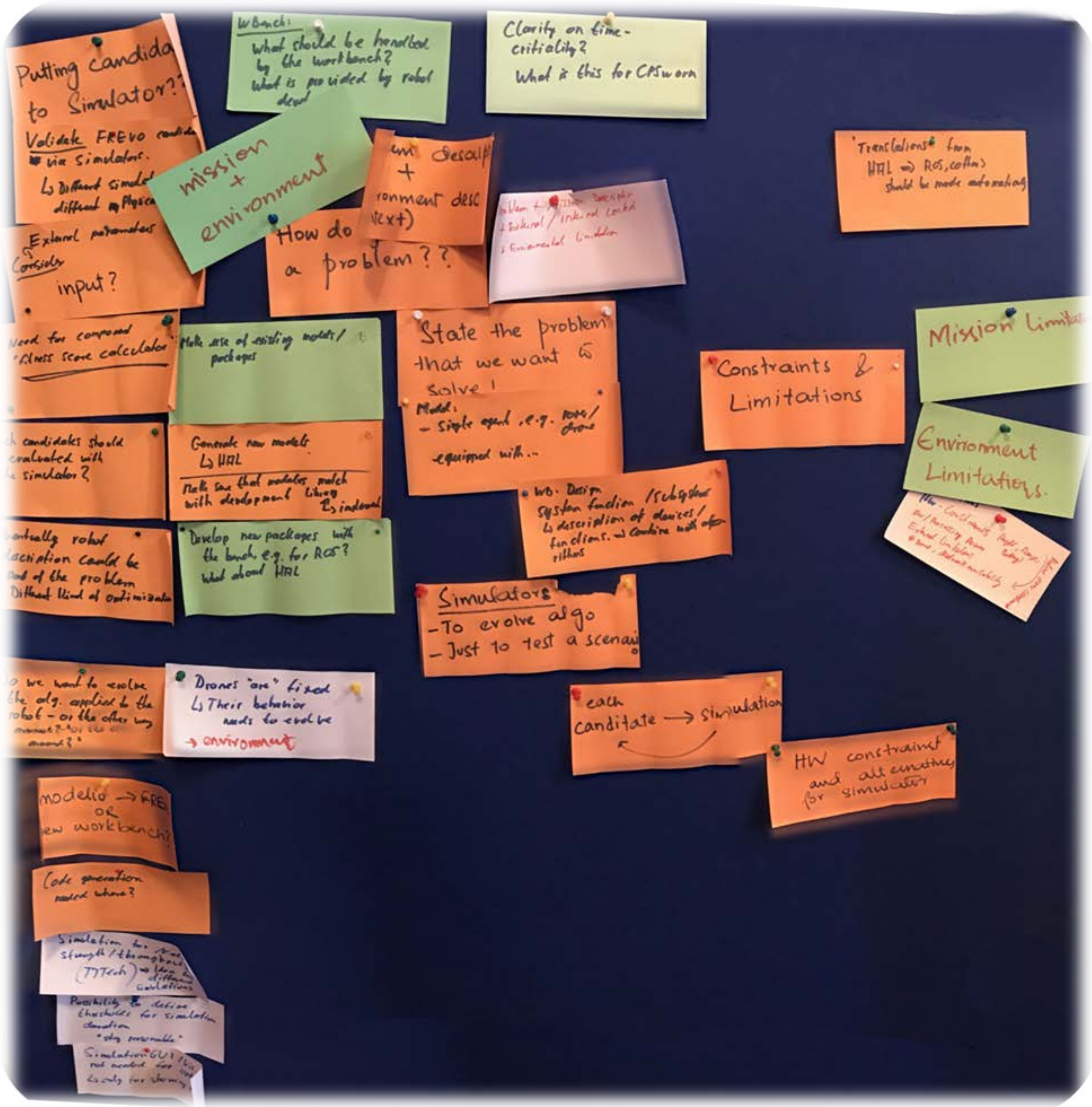

Figure 9: Results on mission and environmental modelling.

Figure 10: Identified needs for modelling.

5.4 Roles

Alongside the different needs and activities, first ideas for roles were identified. Looking at the engineering process from a holistic workbench perspective instead individual tasks each partner was specialized in, it turned out that different activities in a development team could be related to roles. Figure 11 shows an initial set of roles that were named during the session. Later processing of the results applied a higher granularity of the differentiation (cf. Section 6.2).

Figure 11: First collection of identified roles and tasks.

5.5 Libraries

As additional outcome, hidden knowledge such as assumptions that a developer needs to know contextual information about the domain in which he realizes a scenario become visible. Figure 12 shows the outcome of the discussion. For example, partners stated that in a flying drone scenario, you need to know about the battery lifetime of the used drones when planning paths and regulations where flying is allowed. Furthermore, e.g., in a logistics scenario, there are certain laws and regulations that apply when humans may appear in areas where autonomous drones operate. All these kinds of knowledge about technical limitations, dependencies as well as legal aspects are currently provided by the experience of the domain and development expert. The named aspects were taken as input for the knowledge area of the CPSwarm Workbench, namely the collection of available libraries. It turned out that these can be technology-oriented but also provide knowledge about the domain that is important when creating an application concept. Furthermore, knowledge about safety and security measures needs to be made available to the development team.

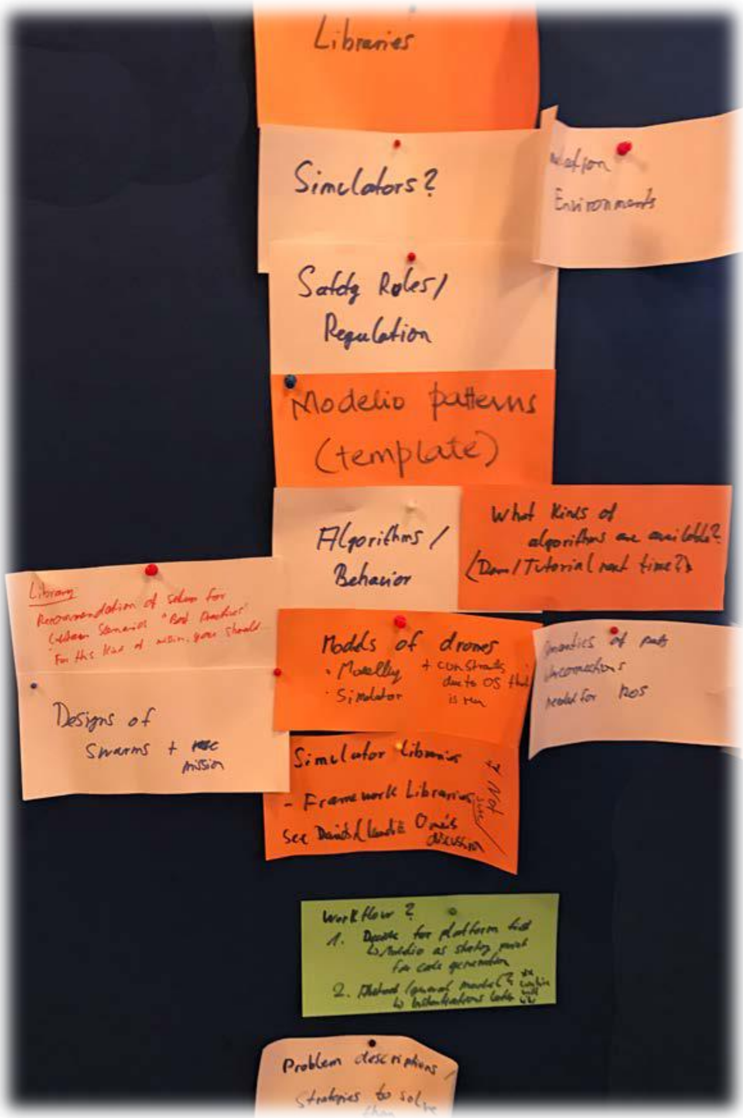

Figure 12:First collection of expert knowledge in libraries.

6 Use Case Analysis

The next step in the engineering process was to analyse all the data gathered in the brainstorming session and convert it into a structured use-case analysis that includes a first overview of involved roles, actors, communication flows and actions performed by each role. Thus, the collected notes were regrouped in such a way that they were clustered and assigned to roles. These were either identified and named during the brainstorming session already or were extracted from sets of actions that were clustered by similarity as shown in Figure 13.

Figure 13 – Structuring actions and deriving of roles.

The aim of this section is to describe various aspects of the use case analysis. Starting with deriving working steps within a design and development team for swarm applications (cf. Section 6.1), stakeholders and relevant actors were identified and are described in Section 6.2. Before providing the actual use case diagrams and descriptions, the communication flow between the relevant actors involved in the design and engineering process is described in Section 0. This section also defines the templates for use case description and diagram which will be used though out this document to describe several use cases. In the end are the detailed descriptions of all the identified use cases.

6.1 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 (cf. Figure 14):

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

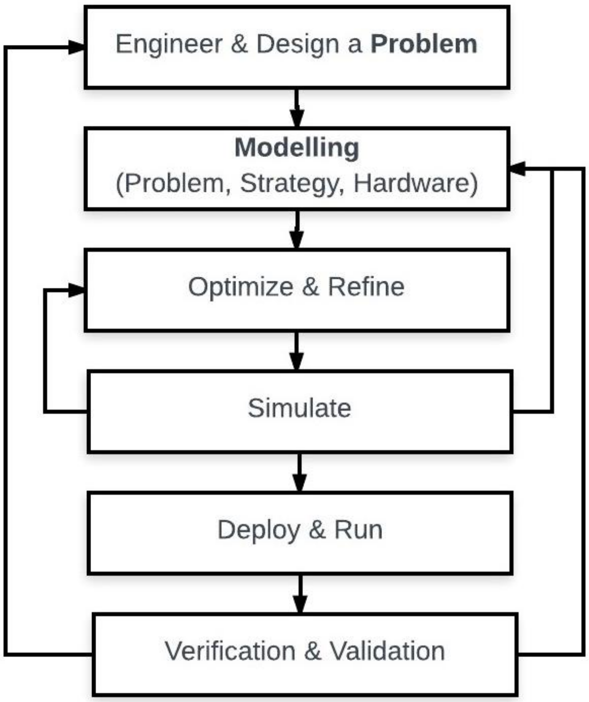

Figure 14 - CPSwarm workbench workflow.

6.2 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 are 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.
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.

6.2.1 Communication Flow between Stakeholders

The previous section explained various stakeholders and their responsibilities. Figure 15 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.

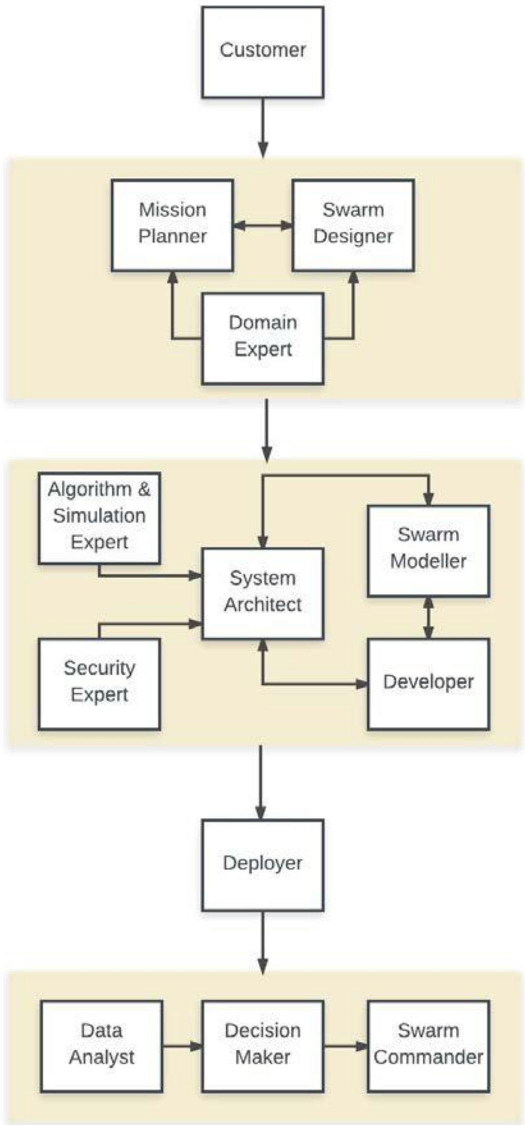

Figure 15 - Communication flow between stakeholders.

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 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.

6.3 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 16 will be used throughout this document to visualize an actor's actions with the system.

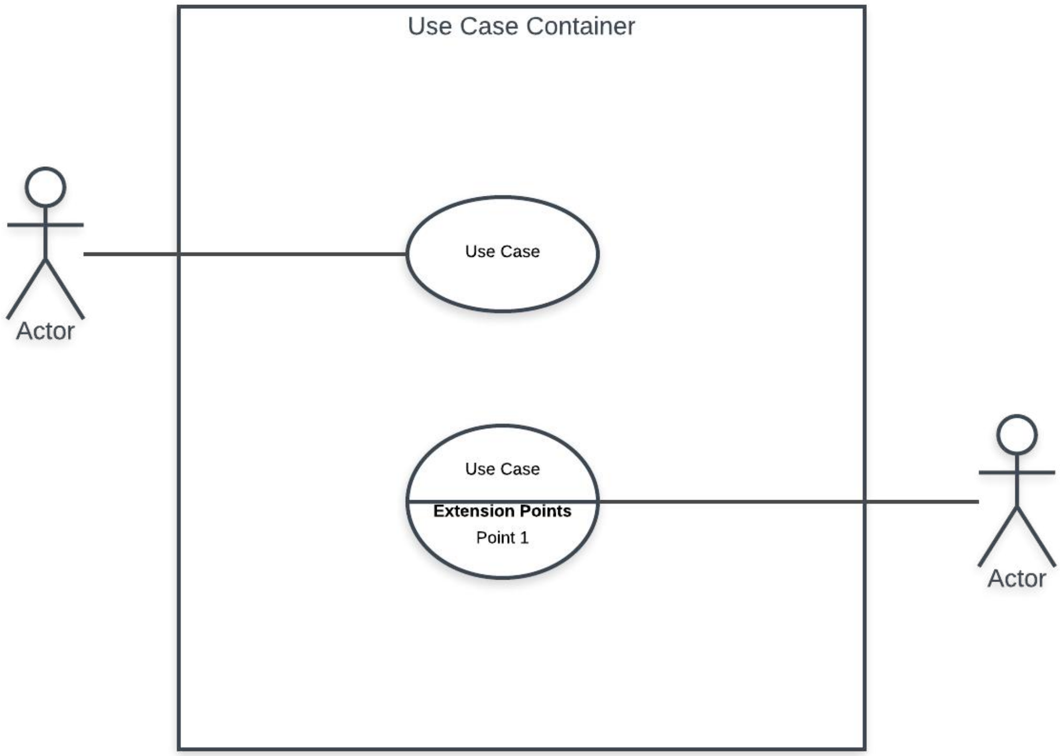

Figure 16 - Use case diagram template

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

6.3.1 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 17 shows all the use cases related to the responsibilities of the Workbench Engineer, followed by their descriptions.

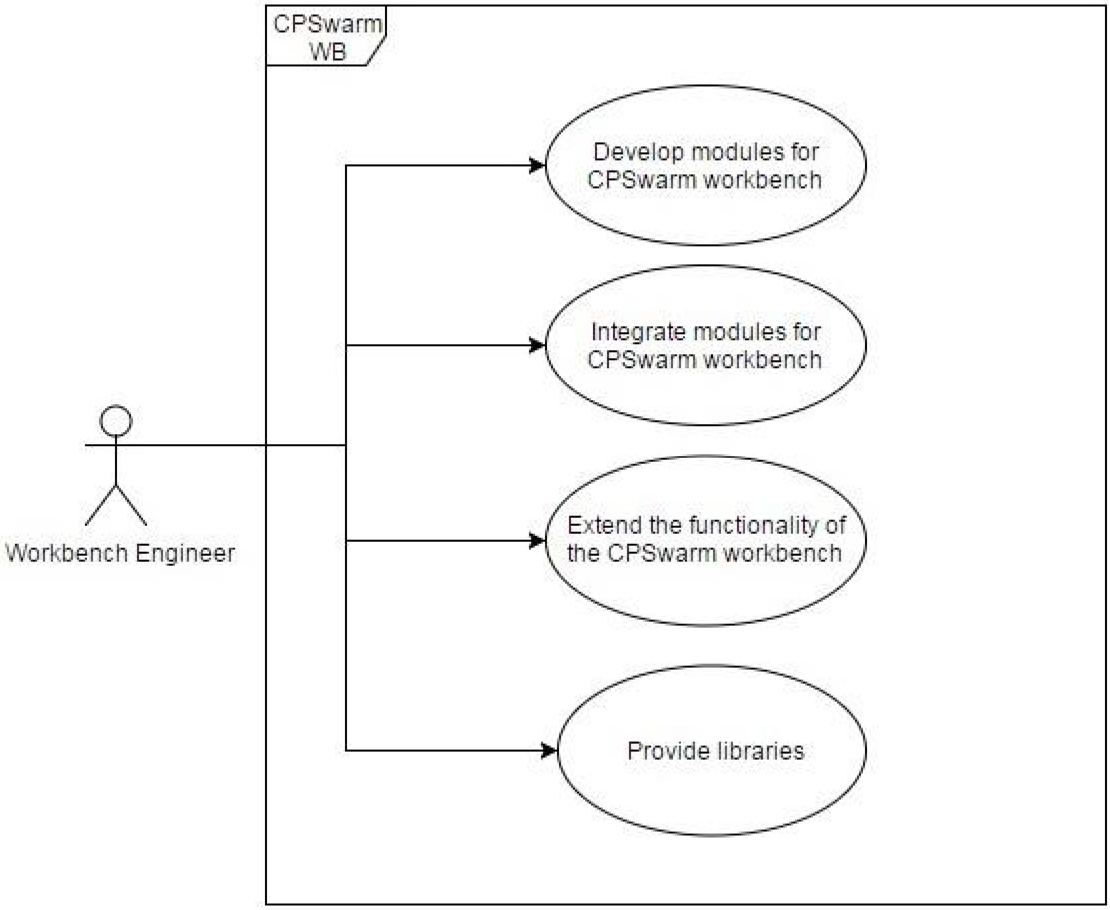

Figure 17 - Workbench Engineer use cases.

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 17
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 17
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 17
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 17
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

6.3.2 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 18 shows all the use cases related to the responsibilities of the mission planner.

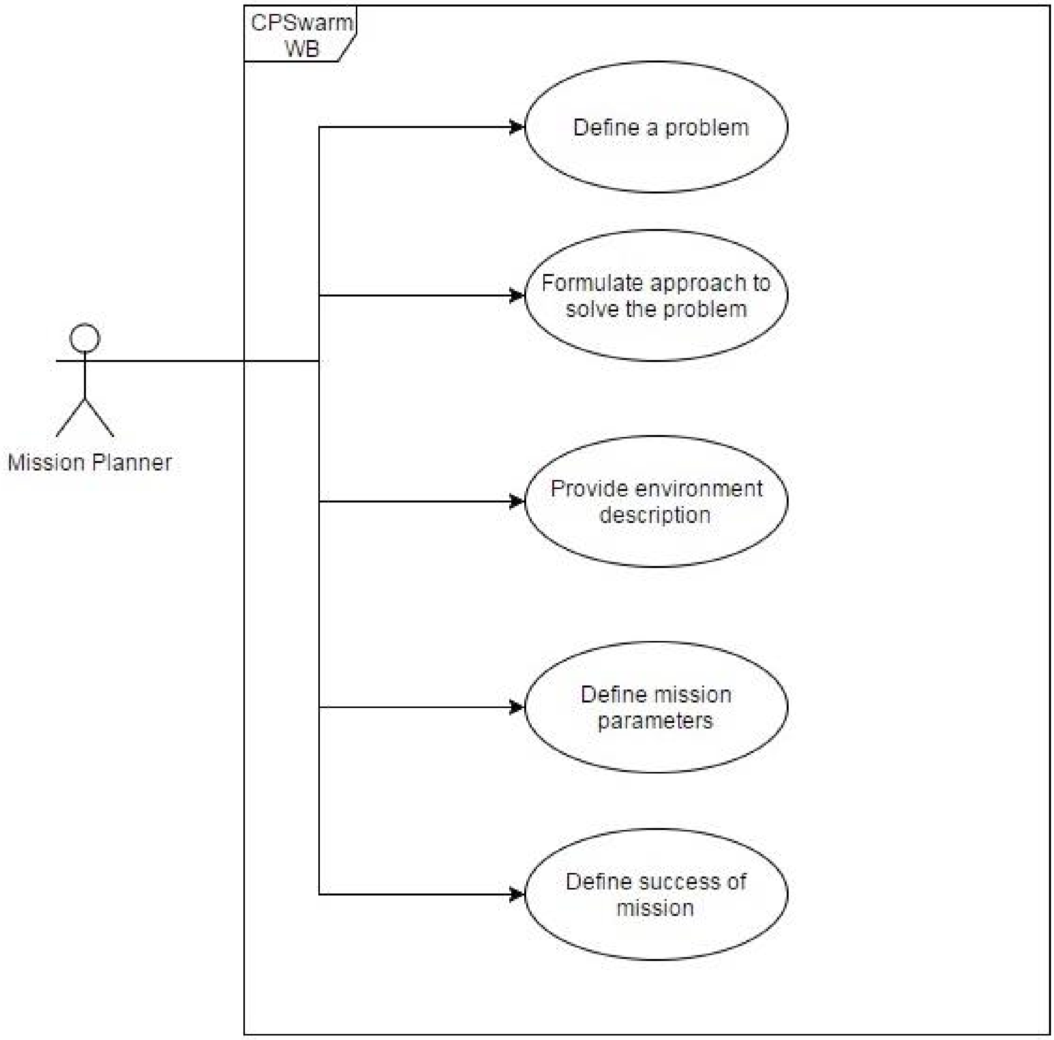

Figure 18 - Mission Planner use cases.

Use Case ID	UC-1.1
Use Case Name	Define a problem
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 18
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 18
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 18
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 18
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 18
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.

6.3.3 Swarm Designer

As described in Table 2, 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 19 shows all the use cases related to the responsibilities of the swarm designer. Each use case is further described below.

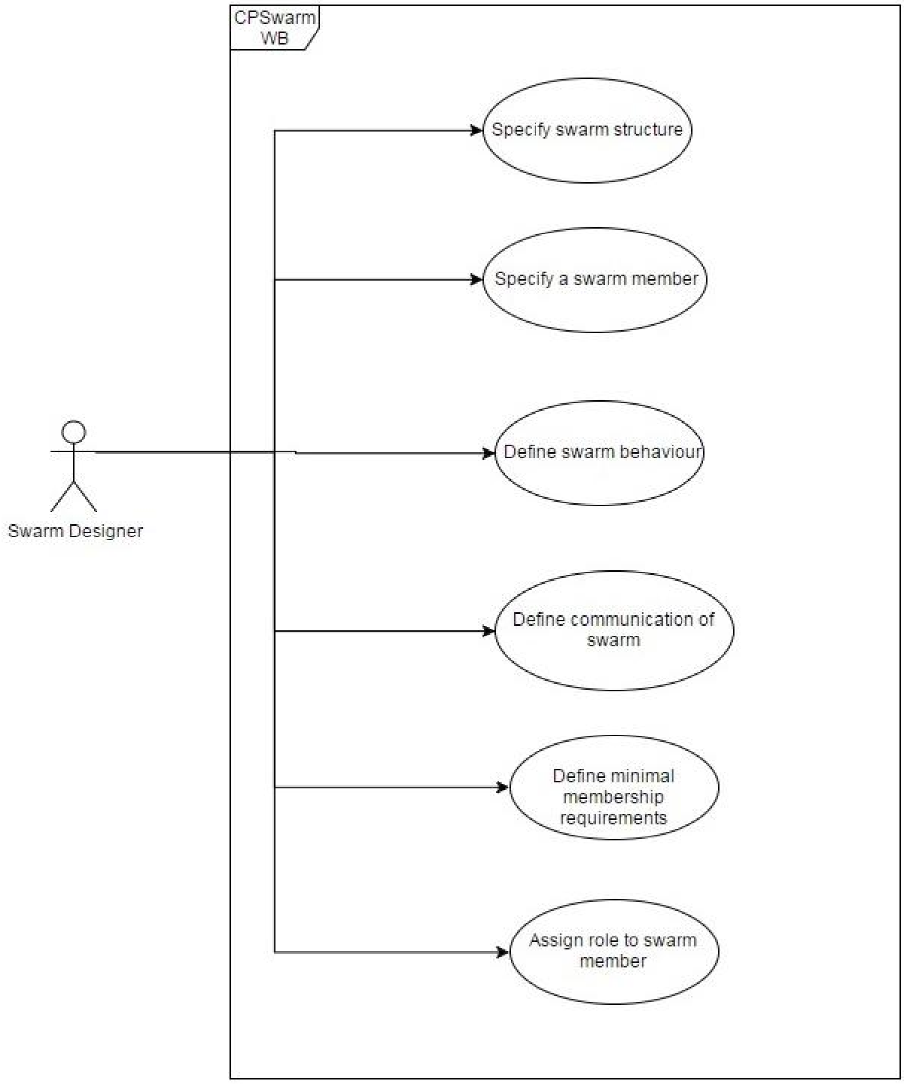

Figure 19 - Swarm Designer use cases.

Use Case ID	UC-2.1
Use Case Name	Specify swarm structure
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 19
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 19
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 19
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 19
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 19
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 19
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.

6.3.4 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 20 shows all the use cases related to the responsibilities of the Domain Expert, followed by their descriptions.

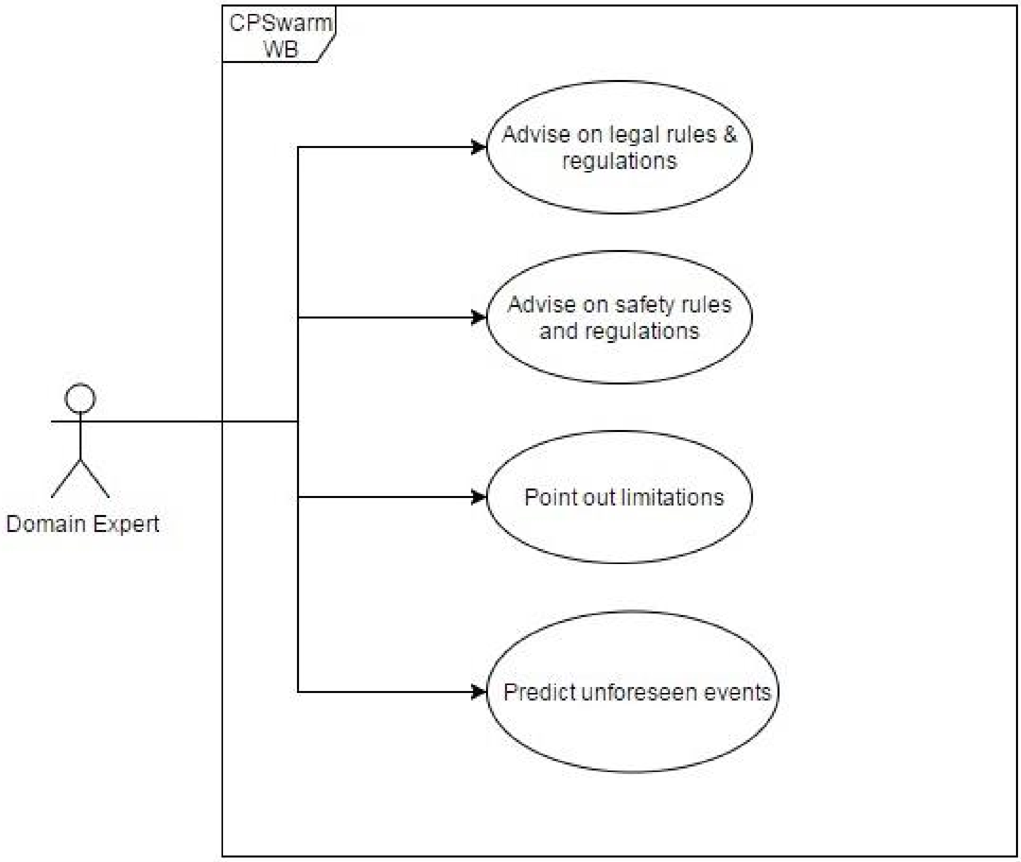

Figure 20 - Domain Expert use cases.

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 20
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 20
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 20
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 20
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

6.3.5 Swarm Modeller

As described in Table 2, 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 21 shows all the use cases related to the responsibilities of the Swarm Modeller, followed by their descriptions.

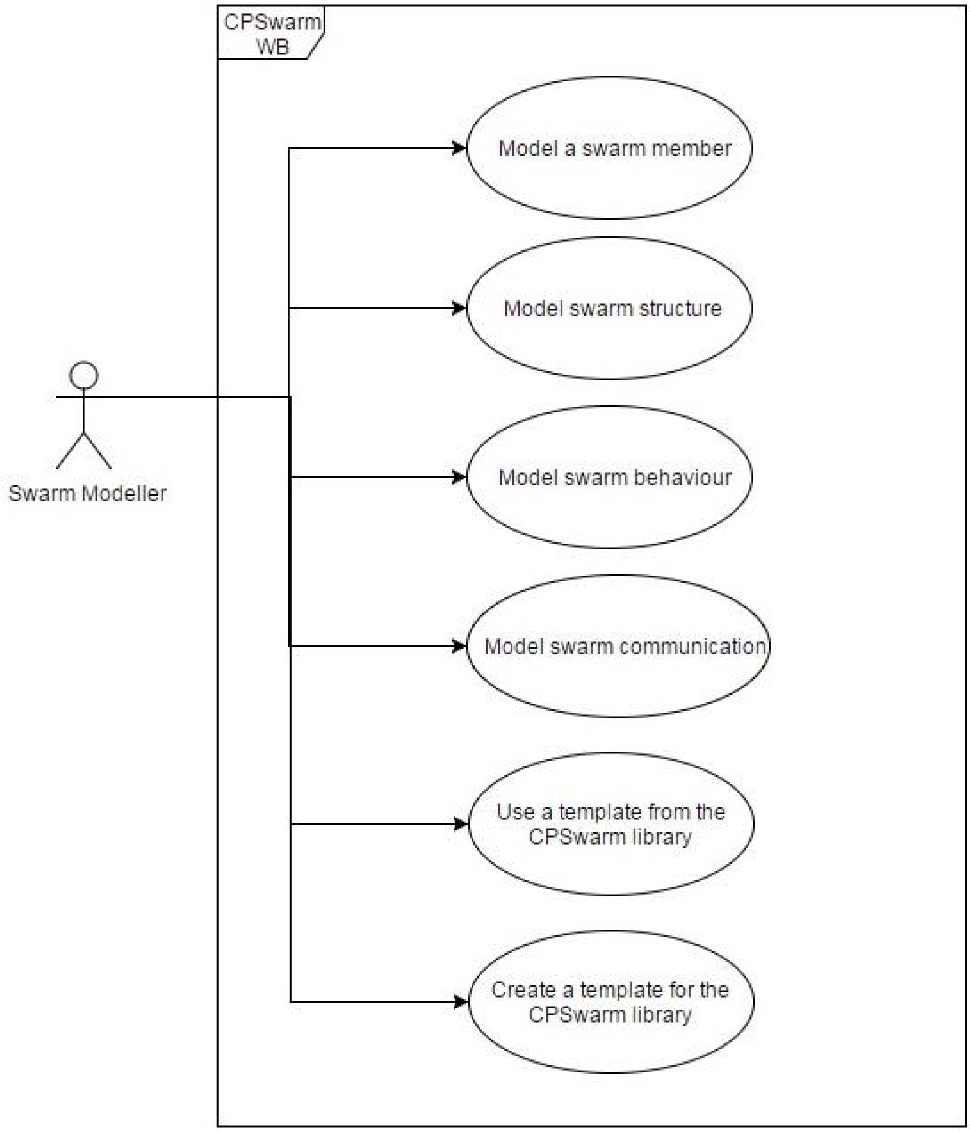

Figure 21 - Swarm Modeller use cases.

Use Case ID	UC-4.1
Use Case Name	Model a swarm member
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 21
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 21
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 21

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 21Figure 20
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 21
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 21
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

6.3.6 Swarm Developer

As described in Table 2, 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 22 shows all the use cases related to the responsibilities of the Swarm Developer, followed by their descriptions.

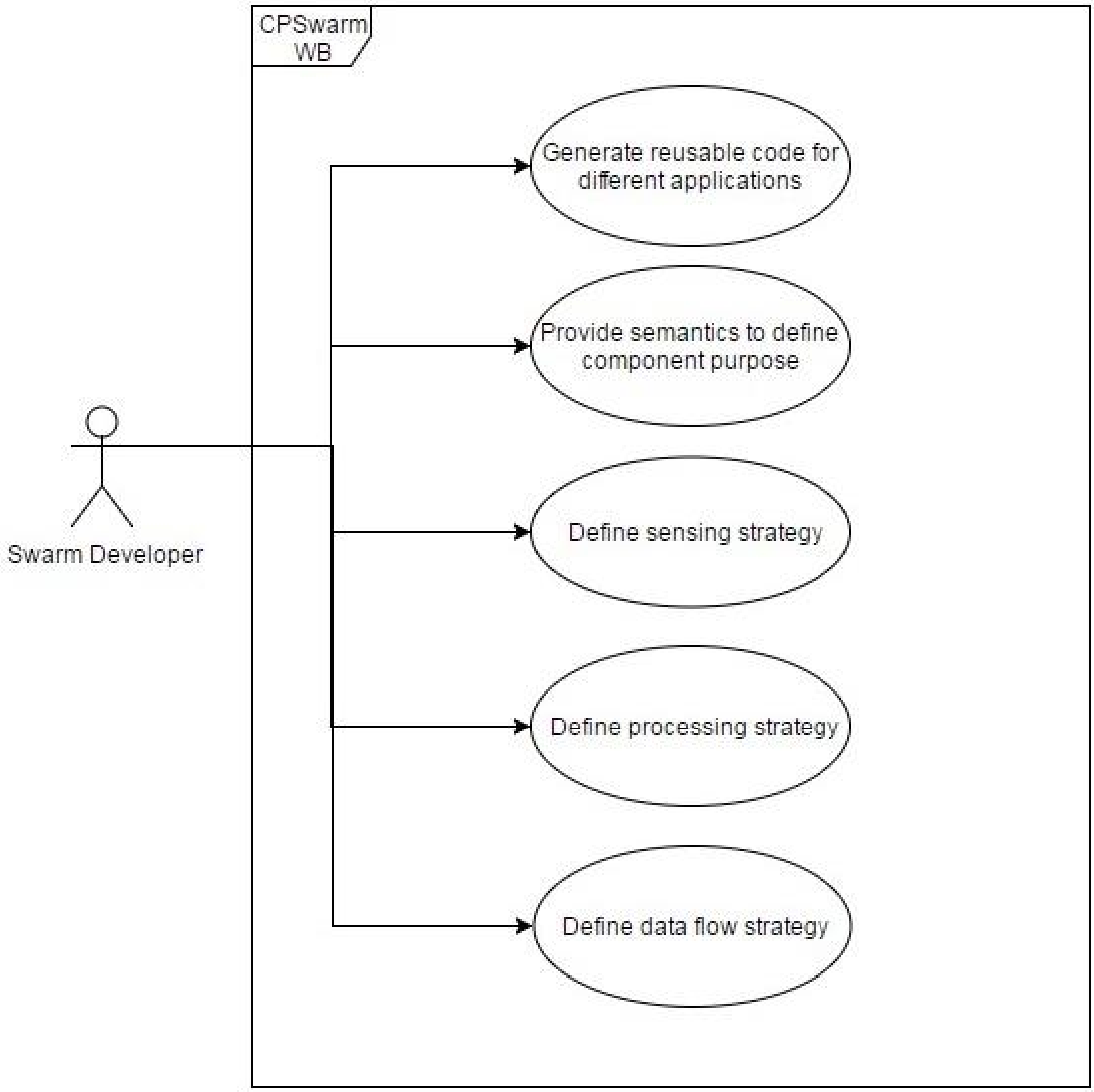

Figure 22 - Swarm Developer use cases.

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 22
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 22
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 22
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 22
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 22
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

6.3.7 Algorithm Optimization & Simulation Expert

As described in Table 2, Algorithm Optimization & Simulation Expert is 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. Figure 23 and Figure 24 show all the use cases related to the responsibilities of the Algorithm Optimization & Simulation Expert, followed by their descriptions.

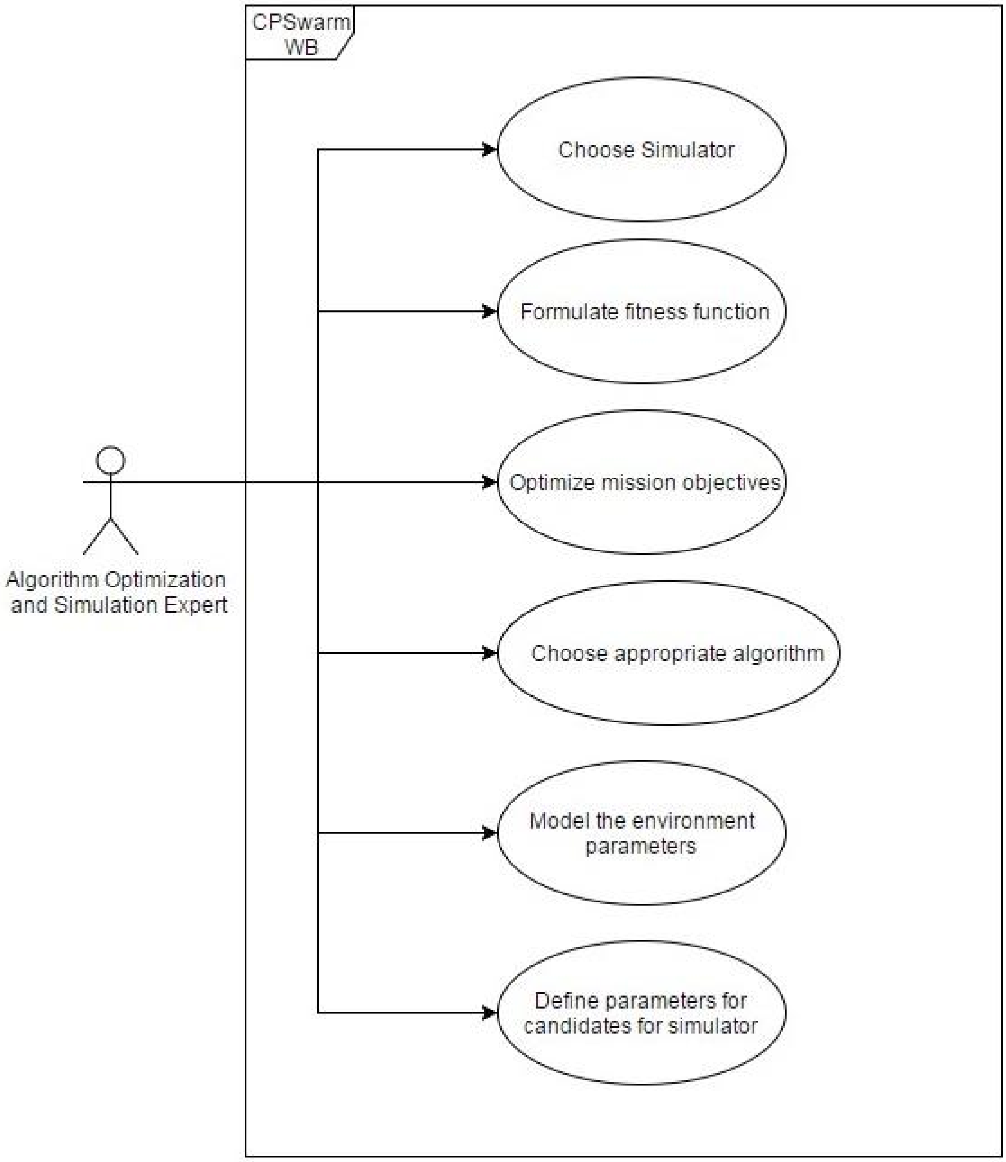

Figure 23 - Algorithm Optimization & Simulation Expert use cases (1).

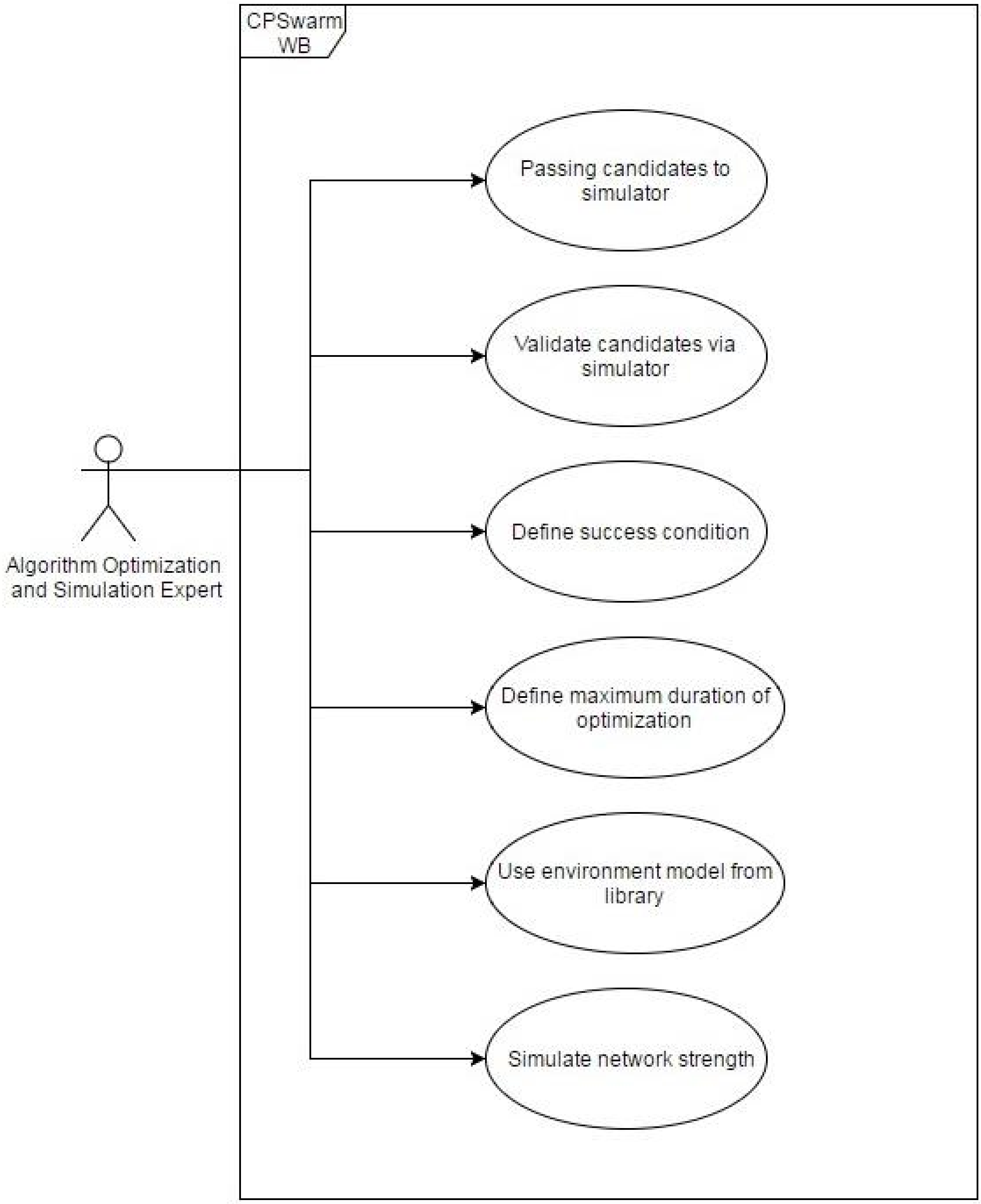

Figure 24 - Algorithm Optimization & Simulation Expert use cases (2).

Use Case ID	UC-6.1
Use Case Name	Choose Simulator
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 23
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 23
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 23
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 23
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 23
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 23
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 24
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 24
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 24
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 24
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 24
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 24
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

6.3.8 Security Expert

As mentioned in Table 2 – Design-related stakeholders, the security expert is responsible for the safety and security aspects of the swarm. 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 25 shows all the use cases related to the security expert. Each use case is further described below.

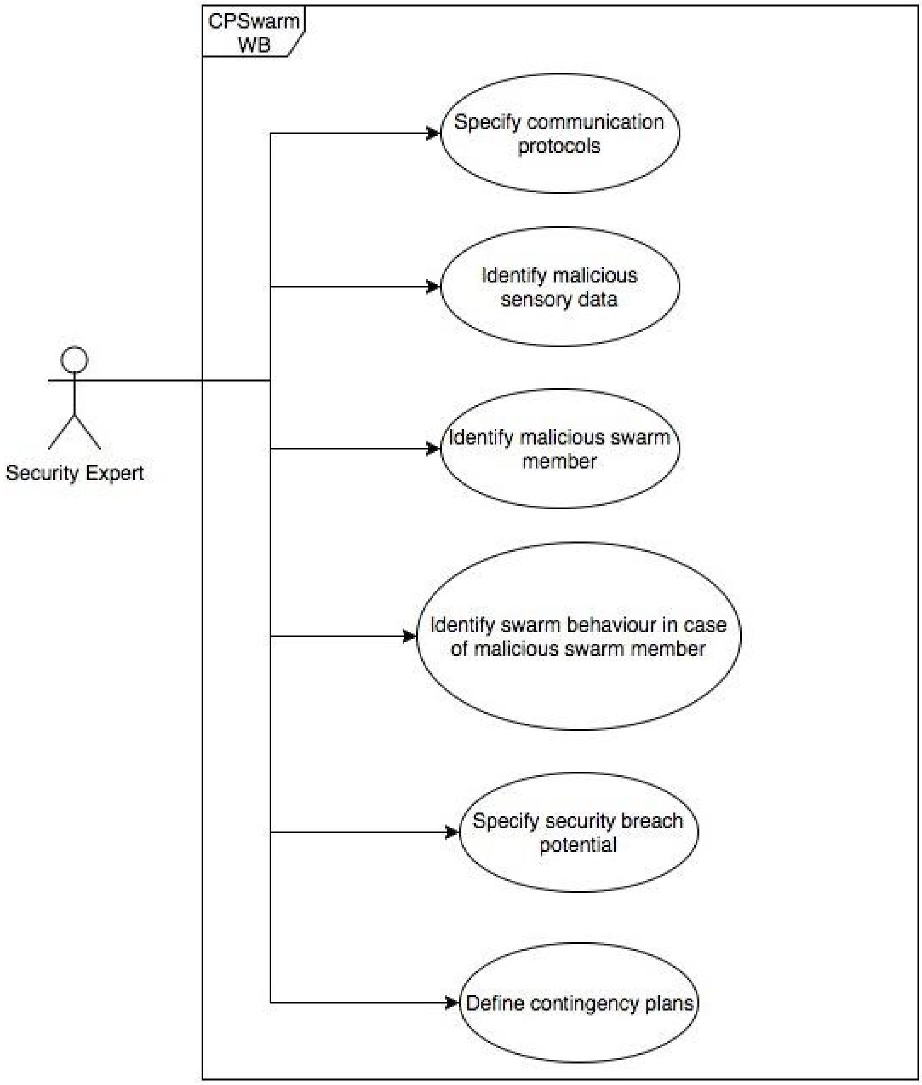

Figure 25 - Security Expert use cases

Use Case ID	UC-7.1
Use Case Name	Specify communication protocol
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 25
Involved Actors	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 25
Involved Actors	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 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 25
Involved Actors	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 25
Involved Actors	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 25
Involved Actors	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 25
Involved Actors	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.

6.3.9 Deployer

As described in Table 2, Deployer is a person or group responsible for deploying the code of the swarm. Figure 26 shows all the use cases related to the responsibilities of the Deployer, followed by their descriptions.

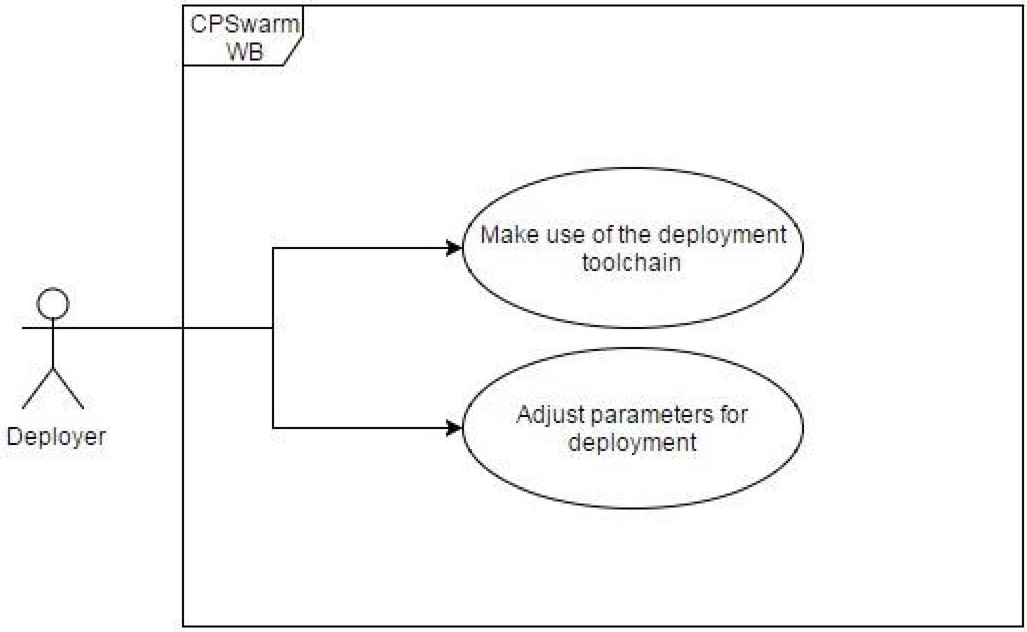

Figure 26 - Deployer use cases.

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 26
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 26
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

6.3.10 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 27 shows all the use cases related to the responsibilities of the Swarm Operator, followed by their descriptions.

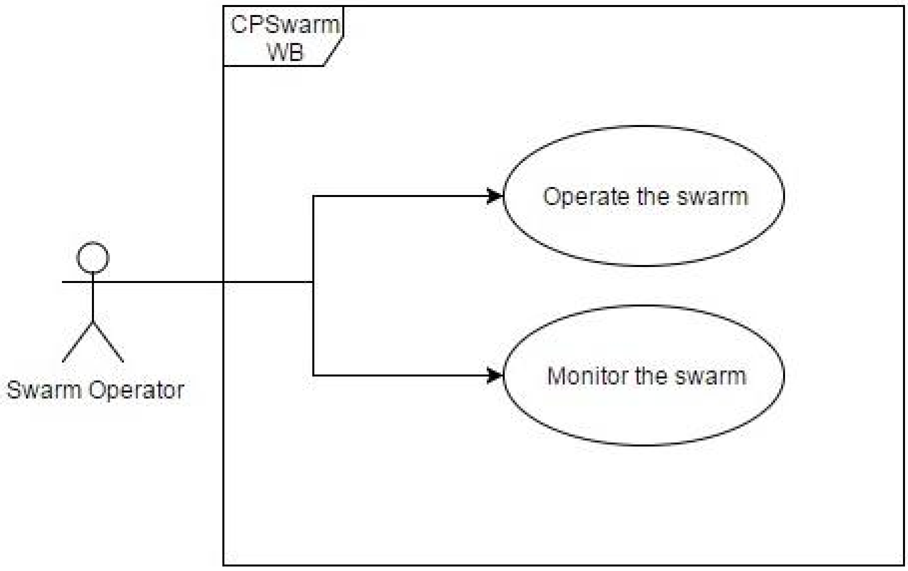

Figure 27 - Swarm Operator use cases.

Use Case ID	UC-9.1
Use Case Name	Operate the swarm
Version	1.0
Author	Sarah Suleri
Use Case Diagram(s)	Figure 27
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 27
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

7 Conclusion

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

One of the objectives 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.

Conclusively, this deliverable documented the iterative process of ideation and concept development in order to identify various stakeholders and generate related use cases. The specification of these use cases and workflows are significant results from this task that will be used as input to subsequent activities of the project.

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