



D2.6 – INITIAL LESSONS LEARNED AND UPDATED REQUIREMENTS REPORT

Deliverable ID	D2.6
Deliverable Title	Initial Lessons Learned and Updated Requirements Report
Work Package	WP2
Dissemination Level	PUBLIC
Version	1.0
Date	2018-07-05
Status	Final
Lead Editor	Sarah Suleri (FRAUNHOFER)
Main Contributors	René Reiners (FRAUNHOFER), Alessandra Bagnato (SOFTEAM), Melanie Schranz (LAKE), Bálint Jánvári, Regina Bíró (SLAB), Gianluca Prato, Davide Conzon, Enrico Ferrera (ISMB), Artiza Elosegui (TTTECH)

Published by the CSPwarm Consortium



Document History

Version	Date	Author(s)	Description
0.1	2018-01-09	Sarah Suleri (FRAUNHOFER)	First Draft with TOC
0.2	2018-02-06	Sarah Suleri (FRAUNHOFER)	Executive Summary Chapter 1,2,3,4,7 inserted Section 5.7,5.9 inserted
		Bálint Jánvári, Regina Bíró (SLAB)	Section 4.3 inserted
		Alessandra Bagnato (SOFTEAM)	Section 5.1, 5.2 inserted
		Melanie Schranz (LAKE)	Section 5.3,5.4 inserted
		Gianluca Prato, Davide Conzon, Enrico Ferrera (ISMB)	Section 5.5,5.6 inserted
		Artiza Elosegui (TTTECH)	Section 5.8 inserted
0.3	2018-07-05	Sarah Suleri (FRAUNHOFER)	Section 4.2 updated to include requirements on security and safety based on the outcome of the Security Workshop in SLAB Updated Section 4.3 Chapter 6 updated
1.0	2018-07-05	Sarah Suleri (FRAUNHOFER)	Minor modifications inserted. Version ready to be submitted to EC

Internal Review History

Review Date	Reviewer	Summary of Comments
2018-02-15	Artiza Elosegui (TTTECH)	Approved
2018-02-16	Omar Morando (DIGISKY)	Approved

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.

WP2 manages and undertakes the work of carrying out the iterative engineering of requirements, which focuses on the engineering process of initial requirements and reengineering after the end of each iteration cycle. The purpose of this work package is thus to maintain a continuous discovery and analysis of user centric requirements, needs and prospects, to be used in the design, development, implementation and validation of the CPSwarm workbench.

The main objective of this deliverable is to describe the reiteration of the initial requirements elicited and documented in D2.3. The goal of this document is to define a list of CPSwarm requirements exploiting the "Volere" approach. These requirements will be continuously updated and refined through an iterative process that will lead to the production of a total of three releases of this document, respectively in Project Months M6, M14 and M26. Additionally, D2.6 also documents lessons learned during design and development of various components of the CPSwarm workbench.

Furthermore, this deliverable formulates the foundation for the validation framework to be specified in D2.8 and updated system architecture analysis and design specifications to be documented in D3.2 in WP3, and later for the remaining technical WPs (WP3 up to WP7), towards the demonstration (WP8).

Table of Contents

1	Introduction.....	5
1.1	Related documents.....	5
2	Approach and Methodology	6
3	Requirements Engineering Approach (FIT)	7
3.1	Volere Requirements Approach	7
3.2	Requirements Management	7
4	CPSwarm Requirement Specification	11
4.1	User Needs	11
4.2	Technical Requirements	12
4.3	Requirements Validation.....	100
5	Lessons Learned.....	101
5.1	Modelling Library.....	101
5.2	Modelling	102
5.3	Optimization	103
5.4	Simulation	104
5.5	Code Generation	105
5.6	Abstraction Layer	105
5.7	Deployment.....	106
5.8	Monitoring.....	106
5.9	Continuous Integration.....	108
6	Design Pattern Library.....	109
7	Conclusion.....	110
	Acronyms.....	111
	List of figures.....	111
	List of tables.....	111
	References.....	111

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 refine initial **user needs** and **technical requirements** identified and described in D2.3 *Initial requirements report*.

This document describes the activities to support the identified workbench workflow, adapting it to the different environments involved in the CPSwarm project and provides a thorough analysis of the requirements. These high-level requirements 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:

- Requirements and user needs reiteration
- Documentation of lessons learned

The results of this deliverable will be continuously updated and refined through an iterative process that will lead to the production of one more release of this document, with the last one planned for project month M26. The development of this deliverable was coordinated by FRAUNHOFER with contribution of SOFTEAM, LAKE, SearchLab, ISMB and TTech. The outcome of this deliverable will be used for deliverable D3.2: *Updated system architecture analysis and design specification* and D2.8: *Validation framework specification*, both due in M18.

1.1 Related documents

ID	Title	Reference	Version	Date
D2.1	Initial Vision Scenarios and Use Case Definition			M4
D2.3	Initial Requirements Report			M6
D3.1	Initial System Architecture Analysis & Design Specification			M6
D3.2	Updated System Architecture Analysis & Design Specification			M18
D2.8	Validation Framework Specification			M18
D2.7	Final Lessons Learned and Requirements Report			M26

2 Approach and Methodology

As depicted in Figure 1, the development cycle for the CPSwarm Workbench starts from the top left with a scenario thinking methodology accompanied by collecting other kinds of input such as related work, documents, standards or available technologies. Once some (partial) understanding of the context has been reached, requirements are derived from it. These requirements, especially in the beginning, take the form of user requirements, i.e. what the user needs from the system. When the system starts to take a concrete shape, these user needs are transformed into technical requirements, i.e. what the system must offer or how the architecture should look like.

In long-term iterations, system design, integration of technologies and knowledge as libraries take place that are then implemented in an incremental manner and later, 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 developments, which, in the end, are validated again. This methodology allows for step-wise knowledge acquisition and development allowing for adjustments alongside conception and development.

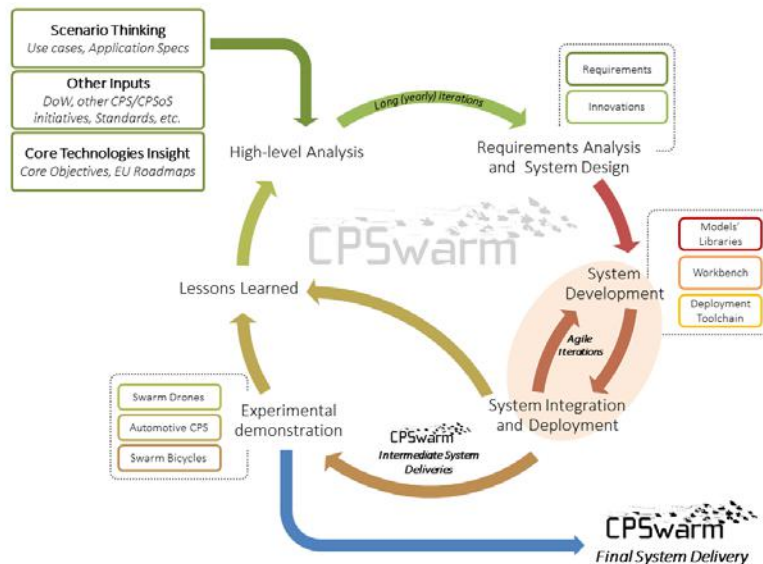


Figure 1: The CPSwarm Workbench development lifecycle

The work reported in this deliverable is located in the top right corner and follows a user centred approach for requirements elicitation. This document also enlists the lessons learned during the system development cycle shown in the bottom right corner. The document's structure is as follows:

- **Chapter 3** describes the Volere requirements approach followed throughout this deliverable for requirement reiteration. This section explains various attributes of the Volere Requirement Shell and additionally, explains the adaptation of this shell used for CPSwarm requirement specification. It also explains details of online support provided for requirement specification and management.
- **Chapter 4** enlists the updated user needs and requirements
- **Chapter 5** describes the lessons learned and recommendations related to each component of the CPSwarm workbench.

3 Requirements Engineering Approach (FIT)

CPSwarm is using the Volere Requirements approach described by Robertson and Robertson (cf. Ref. [9] [10] [11]). Volere is a proven and widely used general-purpose approach to requirements elicitation, including both the process of eliciting requirements as well as the format for representing them. Section 3.1 provides an overview of key elements of the Volere approach.

3.1 Volere Requirements Approach

The Volere requirements approach is described by Robertson and Robertson (cf. Ref [9] [10]). There is a website dedicated to the Volere approach as well: <http://www.volere.co.uk/>. One of the various resources available on this site is the "Volere Requirements Specification Template" (cf. Ref [11]) known as the "Requirements Shell". This format is further explained in subsection 3.1.1.

3.1.1 Requirements Shell

Figure 2 reproduces the Volere "Requirements Shell" by Robertson and Robertson (cf. Ref [11]). While the "Requirements Shell" mimics an index card, it is meant as the definition of a representational format that should be used with appropriate technical support for authoring requirements.

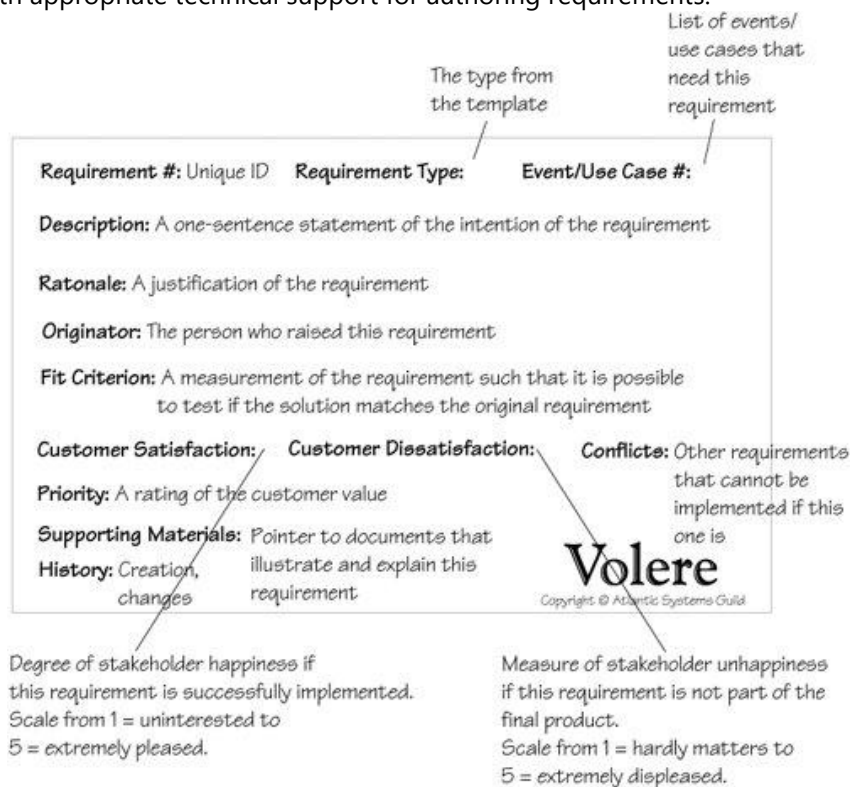


Figure 2: The Volere "Requirements Shell" for representing atomic requirements [11]

3.2 Requirements Management

For the creation and management of information elements of a design process, a number of different approaches have been suggested by Stufflebeam et al. [12] and Penna et al. [7]. In the authors' experience most tools that go beyond Microsoft Word and Excel have little prospect of being used on a broad basis among a heterogeneous group of partners in international R&D projects. While MS Word and Excel are certainly adequate for representing a set of user needs or requirements, they have not proven effective in sustainably supporting a continuous and iterative design process.

As mentioned in D2.3, we used GitLab Issue Tracker for requirement specification and management. For the initial specification of user needs and requirements with limited attributes, the GitLab Issue tracker sufficed but for the reiteration of user needs and requirements, the GitLab could not support all the attributes of the Volere requirements shell. Therefore, we changed the platform to JIRA for requirement specification and management. The CPSwarm JIRA project space can be found at the following address and is hosted by Fraunhofer:

<https://jira.fit.fraunhofer.de/jira/projects/CRD/issues/>

Two general issue types are currently available in the JIRA space of the CPSwarm Requirements and Development project.

The first is a **“User Need”** that documents user stories based on the information acquired from the requirement engineering workshops conducted with our application partners. The creation dialog for user needs in JIRA is shown in Figure 3.

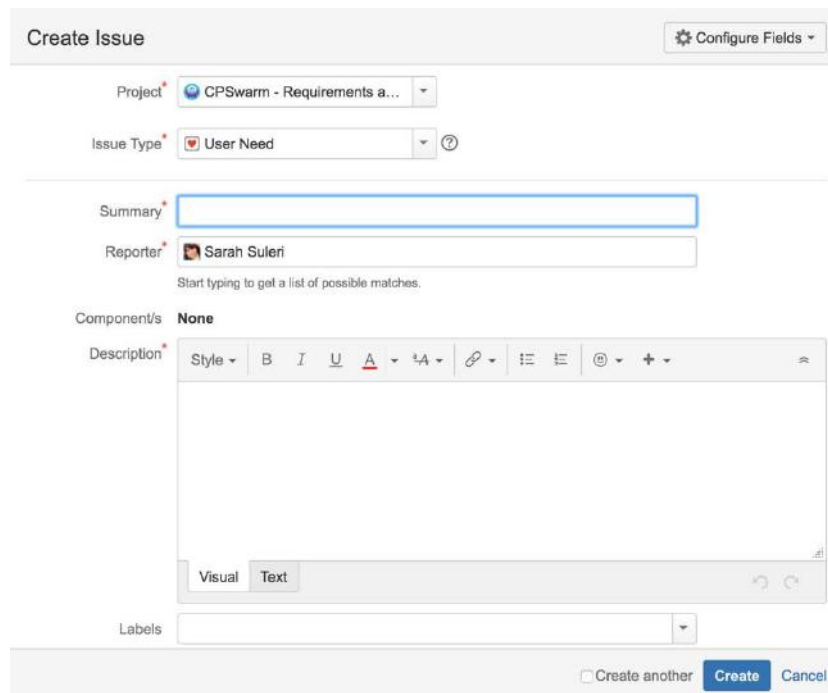


Figure 3: Screenshot of the user need creation dialog in CPSwarm

The second is a **“Volere Requirement”**, which is used to describe various capabilities of different identified components of the workbench. This type of requirements also define the data flow between these components. Figure 4 shows a screenshot of the Volere Requirement creation dialog in JIRA.

Create Issue Configure Fields

Project: CPSwarm - Requirements a...

Issue Type: Volere Requirement

Requirement Type: Functional | None

Event / Use Case: [Empty]

Summary: [Empty]

Description: [Rich text editor with toolbar]

Rationale: [Empty]

Fit Criterion: [Empty]

Customer Satisfaction: None

Customer Dissatisfaction: None

Priority: Minor

Component/s: None

Source: [Empty]

Reporter: Sarah Suleri

Assignee: Automatic

Labels: [Empty]

History: [Empty]

Supporting Material: [Empty]

Linked Issues: mentions

Issue: [Empty]

Create another

Figure 4: Screenshot of the Volere Requirement creation dialog in CPSwarm

A state diagram defining possible states of an issue and appropriate transitions between states has been implemented for both types of issues, "User Need" and "Volere Requirement", in the CPSwarm requirements engineering process. The state diagrams are shown in Figure 5 and Figure 6.

In particular Figure 5 shows that as soon as a user need has been created it is in an "open" state. After an "open" user need has passed the quality check should be set to "quality check passed" state. The state

“duplicate” can be assigned from all states (here the two mentioned above) and means that this user need actually is redundant to another user need, i.e. it duplicates another existing issue.

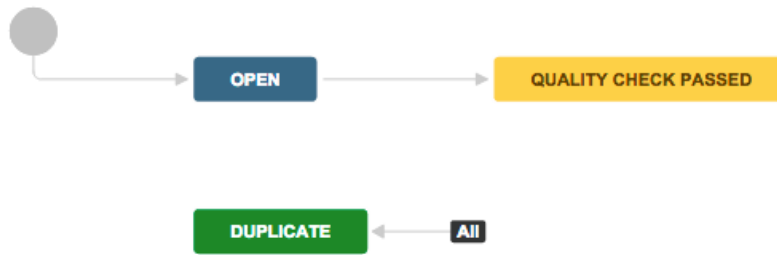


Figure 5: State diagram of issue type “User Need”

Logic depicted in Figure 6 is initially the same as for the previous description. Once the quality check is passed for a requirement, it can become a part of the specification. After the implementation is complete, it acquires the status of “implemented”. After that, it is validated. This three step process is iterative. The figure also shows that once a requirement is open, it can also be rejected based on a legitimate reason.

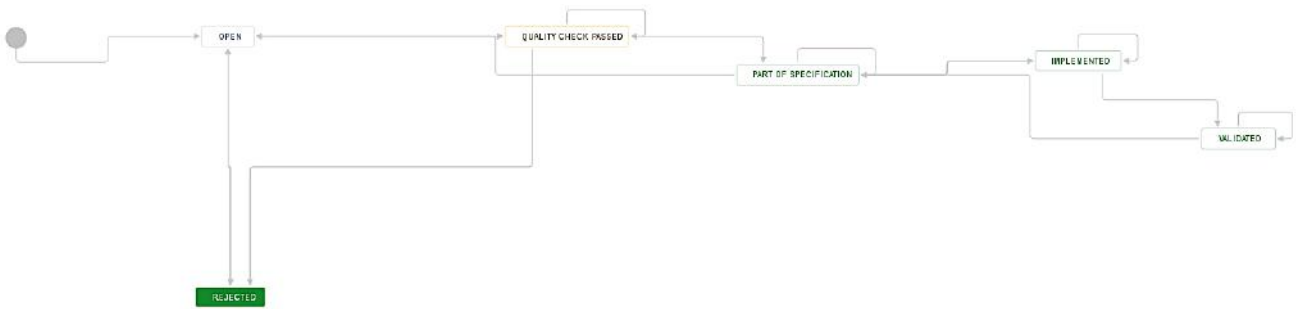


Figure 6: State diagram of issue type “Volere Requirement”

4 CPSwarm Requirement Specification

As mentioned in the previous section, there are two types of requirements described in this document; User Needs and Volere Requirements. User needs are explained from the perspective of various user roles. Whereas, the Volere requirements are described from the perspective of various components of the workbench. The following sections enlist the updated version of these two types of requirements.

4.1 User Needs

Table 1 shows the list of user needs belonging to the user roles identified in D2.3. The **Requirement ID** is a unique identifier for each requirement in the JIRA. Summary as the name describes is the short summary of the user need. **Description** provides a brief detail of the user need.

Table 1: User needs

Requirement ID	Summary	Description
CRD-38	The swarm should consist of self-organizing swarm members	Each swarm member should be autonomous and as a whole the swarm should be able to perform the organization on its own. They should be able to adapt to position, relative actions depending on the area dimensions and sensory information received.
CRD-39	The Swarm Operator should be able to monitor the swarm	Here by monitoring the swarm in play means to: <ul style="list-style-type: none"> • Observe • Receive sensory data • Video feed (optional)
CRD-40	The swarm members should be able to communicate with each other	All the swarm members should be able to pass information (sensory, statistical, positional etc.) to each other
CRD-41	The Swarm Operator should be able to change the mission on the go	Once the swarm is in play under a certain mission, the swarm operator should be able to change the configurations of the mission and the swarm should adapt to these changes. For example, adding a new <ul style="list-style-type: none"> • POI (point of interest) • target • barrier position
CRD-42	Environment conditions should be simulated	Environment conditions such as wind, rain etc. should be simulated
CRD-43	The Mission Planner should be able to configure a mission	For example, for a simple SAR (search and rescue) case, the mission planner should be able to define: <ul style="list-style-type: none"> • an area (dimensions) • target objects • information about swarm members in play
CRD-44	The Mission Planner should be able to start a mission	Once the mission planner has configured a mission successfully, he should be able to start a mission

Requirement ID	Summary	Description
CRD-45	The Mission Planner would like to have a UI to configure a mission	
CRD-46	The Swarm Operator would like to have a UI to monitor the swarm in play	Here by monitoring the swarm in play means to: <ul style="list-style-type: none"> • Observe • Receive sensory data • Video feed (optional)
CRD-47	The swarm can have heterogeneous or a homogeneous composition	The mission planner should be able to include same or different kinds of swarm members in a swarm
CRD-48	The Swarm Designer should be able to define the composition of the swarm	Composition of the swarm means the different types of swarm members the swarm consists of. Also, how many swarm members of each type exist in the swarm
CRD-49	All the swarm members of a swarm should act under only one mission at a time	The mission planner should define one mission for a swarm once and the entire swarm should act accordingly. There should not be a need for the mission planner to define a separate mission or smaller sub-missions for each swarm member
CRD-50	The Mission Planner should be able to add constraints to a mission	For example, maximum altitude, max range starting from the home point, rules and regulations of the area
CRD-51	The Swarm Designer should be able to assign role to swarm member	For example, in a car convoy, the swarm designer assigns a role for a leader vehicle. How is the leader selected or re-selected? <ul style="list-style-type: none"> • Election Algorithm • Manually

4.2 Technical Requirements

Moving one step ahead from user needs, we have visualized the intended system in terms of certain components. Each component has a specific role and certain responsibilities to fulfil. These components all together ensure that all the user needs described in the section 4.1 can be fulfilled. The roles and responsibilities of each component are defined in the form of Volere requirements. Table 2 elaborates various components of the workbench.

Table 2: Workbench component descriptions

Component	Description
Model Library	Repository of different kinds of libraries containing models
Modelling tool	The component responsible for modelling a swarm model
Optimization tool	The component responsible for optimizing swarm algorithm
Optimization Simulator	The component responsible for simulating swarm and calculating fitness score for algorithm optimization

Code Generator	The component responsible for generating deployable code
Deployment tool	The component responsible for deploying the code
Hardware Abstraction Layer	The component responsible for providing abstraction from the technical details of the deployment environment
Monitoring tool	The component responsible for providing real-time feedback about the current state of the swarm

The initial set of technical requirements are documented in D2.3. Below are the details of reiterated Volere requirements.

[CRD-1] The Modelling library will be a collection of different kinds of reusable components			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Critical
Reporter:	Sarah Suleri	Assignee:	Alessandra Bagnato
Labels:	Library		
Issue Links:	Dependency		
	is a dependency of	CRD-2 The Modelling Tool shall be able to us...	Quality Check passed
	is a dependency of	CRD-25 The swarm member library shall contain...	Quality Check passed
	is a dependency of	CRD-26 The swarm member library shall contain...	Quality Check passed
	is a dependency of	CRD-27 The swarm member library shall contain...	Quality Check passed
	is a dependency of	CRD-22 Modelling library shall include a libr...	Quality Check passed
	is a dependency of	CRD-23 The Modelling library shall include a ...	Quality Check passed
	is a dependency of	CRD-24 The Modelling library shall include a...	Quality Check passed
Requirement Type:	Functional		
Event / Use Case:	UC-10.4, UC-4.5, UC-4.6, UC-6.11		
Rationale:	Reusability of pre-existing models to speed up development; Standardization;		
Fit Criterion:	The existence of a collection of reusable components		
Customer Satisfaction:	very high		
Customer Dissatisfaction:	very high		
Source:	Modelling-Optimization Requirements Workshop Bonn (8-9-17)		
History:	Reformulated RE-53		
Supporting Material:	D4.1, D4.4, D2.1		

[CRD-2] The Modelling Tool shall be able to use / reuse models from the Modelling Library			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Critical
Reporter:	Sarah Suleri	Assignee:	Alessandra Bagnato
Labels:	Library, Modelling_Tool		
Issue Links:	Dependency		
	depends on CRD-1 The Modelling library will be a collec... Quality Check passed		
Requirement Type:	Functional		
Event / Use Case:	UC-4.5, UC-4.6		
Rationale:	Integrated toolchain		
Fit Criterion:	User should be able to view, access, use the model library through modelling tool		
Customer Satisfaction:	very high		
Customer Dissatisfaction:	very high		
Source:	Modelling-Optimization Requirements Workshop Bonn (8-9-17)		
History:	Reformulated RE-54		
Supporting Material:	D3.1, D2.3, D5.1		

[CRD-3] The Modelling tool shall be able to model the structure of a swarm member			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Trivial
Reporter:	Sarah Suleri	Assignee:	Alessandra Bagnato
Labels:	Modelling_Tool		
Issue Links:	Dependency		
	is a dependency of	CRD-32	The modelling tool shall be able to mo... Quality Check passed
	is a dependency of	CRD-33	Modelling tool shall be able to model... Quality Check passed
	is a dependency of	CRD-6	The Modelling tool shall be able to mo... Quality Check passed
	is a dependency of	CRD-11	The Modelling tool shall pass the swar... Quality Check passed
	is a dependency of	CRD-15	The Optimization Simulator shall simu... Quality Check passed
	is a dependency of	CRD-7	The Modelling tool shall be able to mo... Quality Check passed
Requirement Type:	Functional		
Event / Use Case:	UC-4.1, UC-4.2		
Rationale:	To give the designer the ability to define the ability of swarm member		
Fit Criterion:	The model of the swarm member structure is editable		
Customer Satisfaction:	neutral		
Customer Dissatisfaction:	low		
Source:	Modelling-Optimization Requirements Workshop Bonn (8-9-17)		
History:	Reformulated RE-55		
Supporting Material:	D4.1, D5.1		

[CRD-4] The Modelling tool shall be able to model the behaviour of a swarm member	
Status:	Quality Check passed

Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Critical
Reporter:	Sarah Suleri	Assignee:	Alessandra Bagnato
Labels:	Modelling_Tool		
Issue Links:	Dependency		
	is a dependency of	CRD-6 The Modelling tool shall be able to mo...	Quality Check passed
	is a dependency of	CRD-11 The Modelling tool shall pass the swar...	Quality Check passed
	is a dependency of	CRD-7 The Modelling tool shall be able to mo...	Quality Check passed
Requirement Type:	Functional		
Event / Use Case:	UC-4.3		
Rationale:	Define individual behaviour to achieve emergent swarm behaviour		
Fit Criterion:	The model of the behaviour of a swarm member is editable		
Customer Satisfaction:	very high		
Customer Dissatisfaction:	very high		
Source:	Modelling-Optimization Requirements Workshop Bonn (8-9-17)		
History:	Reformulated RE-56		
Supporting Material:	D5.1, D4.1		

[CRD-5] The Modelling tool shall be able to model communication between swarm members			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Minor
Reporter:	Sarah Suleri	Assignee:	Alessandra Bagnato
Labels:	Modelling_Tool		
Issue Links:	Dependency		
	is a dependency of	CRD-11 The Modelling tool shall pass the swar...	Quality Check passed
Requirement Type:	Functional		
Event / Use Case:	UC-4.4		
Rationale:	Some tasks require shared information among swarm members		
Fit Criterion:	Swarm member behaviour can exchange information with other swarm members		
Customer Satisfaction:	high		
Customer Dissatisfaction:	neutral		
Source:	Modelling-Optimization Requirements Workshop Bonn (8-9-17)		
History:	Reformulated RE-57		
Supporting Material:	D5.1, D4.1		

[CRD-6] The Modelling tool shall be able to model the composition of a swarm			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Critical
Reporter:	Sarah Suleri	Assignee:	Alessandra Bagnato
Labels:	Modelling_Tool		
Issue Links:	Dependency		
	depends on	CRD-4 The Modelling tool shall be able to mo...	Quality Check passed
	depends on	CRD-3 The Modelling tool shall be able to mo...	Quality Check passed
	is a dependency of	CRD-11 The Modelling tool shall pass the swar...	Quality Check passed
	is a dependency of	CRD-13 The Modelling tool shall pass the swar...	Quality Check passed
	is a dependency of	CRD-15 The Optimization Simulator shall simu...	Quality Check passed
	is a dependency of	CRD-7 The Modelling tool shall be able to mo...	Quality Check passed
Requirement Type:	Functional		
Event / Use Case:	UC-4.2		
Rationale:	It is necessary for optimization, simulation and deployment		
Fit Criterion:	User can build a swarm with a number of different swarm members.		
Customer Satisfaction:	very high		
Customer Dissatisfaction:	very high		
Source:	Modelling-Optimization Requirements Workshop Bonn (8-9-17)		
History:	Reformulated RE-58		
Supporting Material:	D4.4		
[CRD-7] The Modelling tool shall be able to model fitness function to define the goal of the swarm behaviour			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		

Type:	Volere Requirement	Priority:	Minor
Reporter:	Sarah Suleri	Assignee:	Alessandra Bagnato
Labels:	Fitness_Function, Modelling_Tool		
Issue Links:	Dependency		
	depends on	CRD-4 The Modelling tool shall be able to mo...	Quality Check passed
	depends on	CRD-6 The Modelling tool shall be able to mo...	Quality Check passed
	depends on	CRD-3 The Modelling tool shall be able to mo...	Quality Check passed
	is a dependency of	CRD-31 The modelling tool shall contain an e...	Quality Check passed
	is a dependency of	CRD-11 The Modelling tool shall pass the swar...	Quality Check passed
	is a dependency of	CRD-12 The Modelling tool shall pass fitness ...	Quality Check passed
Requirement Type:	Functional		
Event / Use Case:	UC-1.5, UC-4.3, UC-6.2, UC-6.3		
Rationale:	Required to run the optimization process that leads to emergent behaviour		
Fit Criterion:	Mathematically represent the fitness function using the parameters from the swarm members		
Customer Satisfaction:	high		
Customer Dissatisfaction:	neutral		
Source:	Modelling-Optimization Requirements Workshop Bonn (8-9-17)		
History:	Reformulated RE-59		
Supporting Material:	D5.1, D4.4		

[CRD-8] The Modelling tool shall be able to model the environment			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Sarah Suleri	Assignee:	Alessandra Bagnato

Labels:	Modelling_Tool												
Issue Links:	<table border="1"> <thead> <tr> <th colspan="4">Dependency</th> </tr> </thead> <tbody> <tr> <td>is a dependency of</td> <td>CRD-11</td> <td>The Modelling tool shall pass the swar...</td> <td>Quality Check passed</td> </tr> <tr> <td>is a dependency of</td> <td>CRD-16</td> <td>The Optimization Simulator shall simu...</td> <td>Quality Check passed</td> </tr> </tbody> </table>	Dependency				is a dependency of	CRD-11	The Modelling tool shall pass the swar...	Quality Check passed	is a dependency of	CRD-16	The Optimization Simulator shall simu...	Quality Check passed
Dependency													
is a dependency of	CRD-11	The Modelling tool shall pass the swar...	Quality Check passed										
is a dependency of	CRD-16	The Optimization Simulator shall simu...	Quality Check passed										
Requirement Type:	Functional												
Event / Use Case:	UC-1.3, UC-6.11, UC-6.5												
Rationale:	The description of environment is necessary for optimization, simulation, etc.												
Fit Criterion:	User is able to define parameters of the environment												
Customer Satisfaction:	very high												
Customer Dissatisfaction:	high												
Source:	Brainstorming Session Bonn												
History:	RE-61												
Supporting Material:	D4.1												

[CRD-9] The Modelling tool shall pass the end condition of simulation to the Optimization tool											
Status:	Quality Check passed										
Project:	CPSwarm - Requirements and Development										
Type:	Volere Requirement	Priority:	Nice to have								
Reporter:	Sarah Suleri	Assignee:	Alessandra Bagnato								
Labels:	Modelling_Tool, Optimization_Tool										
Issue Links:	<table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th colspan="4">Dependency</th> </tr> </thead> <tbody> <tr> <td>is a dependency of</td> <td>CRD-10</td> <td>The Modelling tool shall pass the envi...</td> <td>Quality Check passed</td> </tr> </tbody> </table>			Dependency				is a dependency of	CRD-10	The Modelling tool shall pass the envi...	Quality Check passed
Dependency											
is a dependency of	CRD-10	The Modelling tool shall pass the envi...	Quality Check passed								
Requirement Type:	Functional										
Event / Use Case:	UC-6.10, UC-6.9										
Rationale:	The simulation must have an end condition, otherwise it runs forever										
Fit Criterion:	Optimization tool receives the end condition from modelling tool										
Customer Satisfaction:	low										
Customer Dissatisfaction:	low										
Source:	Modelling-Optimization Requirements Workshop Bonn (8-9-17)										
History:	Reformulated RE-62										

[CRD-10] The Modelling tool shall pass the environment model to the Optimization tool			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Critical
Reporter:	Sarah Suleri	Assignee:	Alessandra Bagnato
Labels:	Modelling_Tool, Optimization_Tool		
Issue Links:	<p style="text-align: center;">Dependency</p> <p>depends on CRD-9 The Modelling tool shall pass the end ... Quality Check passed</p>		
Requirement Type:	Functional		
Event / Use Case:	UC-1.3, UC-1.4, UC-6.11, UC-6.5, UC-6.6		
Rationale:	Environment model must be known to simulation tool for simulation		
Fit Criterion:	Simulation tool receives the environment model from modelling tool		
Customer Satisfaction:	very high		
Customer Dissatisfaction:	very high		
History:	Reformulated RE-63		
Supporting Material:	D3.1		

[CRD-11] The Modelling tool shall pass the swarm model to the Optimization tool																								
Status:	Quality Check passed																							
Project:	CPSwarm - Requirements and Development																							
Type:	Volere Requirement	Priority:	Critical																					
Reporter:	Sarah Suleri	Assignee:	Alessandra Bagnato																					
Labels:	Modelling_Tool, Optimization_Tool																							
Issue Links:	<table border="1"> <thead> <tr> <th colspan="3">Dependency</th> </tr> </thead> <tbody> <tr> <td>depends on CRD-4</td> <td>The Modelling tool shall be able to mo...</td> <td>Quality Check passed</td> </tr> <tr> <td>depends on CRD-6</td> <td>The Modelling tool shall be able to mo...</td> <td>Quality Check passed</td> </tr> <tr> <td>depends on CRD-8</td> <td>The Modelling tool shall be able to mo...</td> <td>Quality Check passed</td> </tr> <tr> <td>depends on CRD-5</td> <td>The Modelling tool shall be able to mo...</td> <td>Quality Check passed</td> </tr> <tr> <td>depends on CRD-7</td> <td>The Modelling tool shall be able to mo...</td> <td>Quality Check passed</td> </tr> <tr> <td>depends on CRD-3</td> <td>The Modelling tool shall be able to mo...</td> <td>Quality Check passed</td> </tr> </tbody> </table>			Dependency			depends on CRD-4	The Modelling tool shall be able to mo...	Quality Check passed	depends on CRD-6	The Modelling tool shall be able to mo...	Quality Check passed	depends on CRD-8	The Modelling tool shall be able to mo...	Quality Check passed	depends on CRD-5	The Modelling tool shall be able to mo...	Quality Check passed	depends on CRD-7	The Modelling tool shall be able to mo...	Quality Check passed	depends on CRD-3	The Modelling tool shall be able to mo...	Quality Check passed
Dependency																								
depends on CRD-4	The Modelling tool shall be able to mo...	Quality Check passed																						
depends on CRD-6	The Modelling tool shall be able to mo...	Quality Check passed																						
depends on CRD-8	The Modelling tool shall be able to mo...	Quality Check passed																						
depends on CRD-5	The Modelling tool shall be able to mo...	Quality Check passed																						
depends on CRD-7	The Modelling tool shall be able to mo...	Quality Check passed																						
depends on CRD-3	The Modelling tool shall be able to mo...	Quality Check passed																						
Requirement Type:	Functional																							
Rationale:	The swarm model is necessary for optimization																							
Fit Criterion:	The optimization tool receives the swarm model which includes the structure, behaviour and communication model of an individual member of the swarm as well as the entire swarm.																							
Customer Satisfaction:	very high																							
Customer Dissatisfaction:	very high																							
Source:	Brainstorming session Bonn																							
History:	RE-64																							
Supporting Material:	D3.1, D2.3																							

[CRD-12] The Modelling tool shall pass fitness function to the Optimization Tool			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Sarah Suleri	Assignee:	Alessandra Bagnato
Labels:	Fitness_Function, Modelling_Tool, Optimization_Simulator		
Issue Links:	<p style="text-align: center;">Dependency</p> <p>depends on CRD-7 The Modelling tool shall be able to mo... Quality Check passed</p>		
Requirement Type:	Functional		
Event / Use Case:	UC-1.5, UC-6.2, UC-6.3		
Rationale:	Required to run the optimization process that leads to emergent behaviour		
Fit Criterion:	Mathematically represent the fitness function using the parameters from the swarm members		
Customer Satisfaction:	high		
Customer Dissatisfaction:	high		
Source:	Modelling-Optimization Requirements Workshop Bonn (8-9-17)		
History:	Reformulated RE-65		
Supporting Material:	D3.1, D2.3		

[CRD-13] The Modelling tool shall pass the swarm composition to the Optimization Tool			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Sarah Suleri	Assignee:	Alessandra Bagnato
Labels:	Modelling_Tool, Optimization_Simulator		
Issue Links:	<p style="text-align: center;">Dependency</p> <p>depends on CRD-6 The Modelling tool shall be able to mo... Quality Check passed</p>		
Requirement Type:	Functional		
Event / Use Case:	UC-2.1, UC-4.2		
Rationale:	The swarm composition is necessary for simulation		
Fit Criterion:	The simulator has the information regarding the swarm composition		
Customer Satisfaction:	high		
Customer Dissatisfaction:	high		
Source:	Modelling-Optimization Requirements Workshop Bonn (8-9-17)		
History:	Reformulated RE-66		
Supporting Material:	D3.1		

[CRD-14] Optimization tool shall pass operational commands to the Optimization Simulator			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Sarah Suleri	Assignee:	Melanie Schranz
Labels:	Optimization_Simulator, Optimization_Tool		
Requirement Type:	Functional		
Event / Use Case:	UC-6.6		
Rationale:	Optimization Tool control the simulated swarm members in simulator, as well as the simulator		
Fit Criterion:	The simulated swarm members behave as indicated by optimization tool		
Customer Satisfaction:	high		
Customer Dissatisfaction:	high		
Source:	Modelling-Optimization Requirements Workshop Bonn (8-9-17)		
History:	Reformulated RE-68		
Supporting Material:	D3.1		

[CRD-15] The Optimization Simulator shall simulate swarm composition, swarm member structure			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Sarah Suleri	Assignee:	Melanie Schranz
Labels:	Optimization_Simulator		
Issue Links:	Dependency		
	depends on CRD-6 The Modelling tool shall be able to mo...		Quality Check passed
	depends on CRD-3 The Modelling tool shall be able to mo...		Quality Check passed
Requirement Type:	Functional		
Event / Use Case:	UC-6.1, UC-6.12, UC-6.7, UC-6.8		
Rationale:	Fitness score needs simulation for sensor input and parameters to be calculated		
Fit Criterion:	The simulator is able to simulate the scenario as per the given information		
Customer Satisfaction:	high		
Customer Dissatisfaction:	high		
Source:	Modelling-Optimization Requirements Workshop Bonn (8-9-17)		
History:	Reformulated RE-69		
Supporting Material:	D3.1		

[CRD-16] The Optimization Simulator shall simulate environment model			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Sarah Suleri	Assignee:	Melanie Schranz
Labels:	Optimization_Simulator		
Issue Links:	<p style="text-align: center;">Dependency</p> <p>depends on CRD-8 The Modelling tool shall be able to mo... Quality Check passed</p>		
Requirement Type:	Functional		
Event / Use Case:	UC-1.3, UC-6.11, UC-6.5		
Rationale:	Simulate details of the environment		
Fit Criterion:	The effect of the environment is reflected on the swarm members in the simulation		
Customer Satisfaction:	high		
Customer Dissatisfaction:	high		
Source:	Brainstorming session Bonn		
History:	RE-70		
Supporting Material:	D3.1, D6.1		

[CRD-17] The Optimization Simulator shall calculate fitness score for each simulation			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Sarah Suleri	Assignee:	Melanie Schranz
Labels:	Optimization_Simulator		
Issue Links:	Dependency		
	is a dependency of	CRD-20 Optimization Tool shall optimize the ...	Quality Check passed
	is a dependency of	CRD-18 The Optimization Simulator shall pass...	Quality Check passed
	is a dependency of	CRD-19 The Optimization Simulator shall pass...	Quality Check passed
Requirement Type:	Functional		
Event / Use Case:	UC-1.5, UC-6.2, UC-6.3		
Rationale:	The fitness of candidate controller needs to be rated by fitness score		
Fit Criterion:	A fitness score is return at the end of simulation from simulator		
Customer Satisfaction:	high		
Customer Dissatisfaction:	high		
Source:	Brainstorming session Bonn		
History:	RE-72		
Supporting Material:	D3.1, D6.1		

[CRD-18] The Optimization Simulator shall pass the fitness score to the Optimization tool			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Sarah Suleri	Assignee:	Melanie Schranz
Labels:	Optimization_Simulator		
Issue Links:	Dependency		
	depends on	CRD-17	The Optimization Simulator shall calc... Quality Check passed
	is a dependency of	CRD-20	Optimization Tool shall optimize the ... Quality Check passed
Requirement Type:	Functional		
Event / Use Case:	UC-1.5, UC-6.2, UC-6.3		
Rationale:	The fitness of candidate controller needs to be ranked by the optimization tool		
Fit Criterion:	A fitness score is received by the optimization tool at the end of simulation		
Customer Satisfaction:	high		
Customer Dissatisfaction:	high		
Source:	Brainstorming session Bonn		
History:	RE-73		
Supporting Material:	D3.1		

[CRD-19] The Optimization Simulator shall pass the sensor data of each swarm member back to optimization tool			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Sarah Suleri	Assignee:	Melanie Schranz
Labels:	Optimization_Simulator, Optimization_Tool		
Issue Links:	<div style="background-color: #f4b084; text-align: center; padding: 5px;">Dependency</div> depends on CRD-17 The Optimization Simulator shall calc... Quality Check passed		
Requirement Type:	Functional		
Event / Use Case:	UC-6.8		
Rationale:	Optimization Tool must have the sensor data to compute operational commands		
Fit Criterion:	Sensor data are received in optimization tool		
Customer Satisfaction:	high		
Customer Dissatisfaction:	high		
Source:	Modelling-Optimization Requirements Workshop Bonn (8-9-17)		
History:	Reformulated RE-74		
Supporting Material:	D3.1, D6.1		

[CRD-20] Optimization Tool shall optimize the algorithm according to the fitness score			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Critical
Reporter:	Sarah Suleri	Assignee:	Melanie Schranz
Labels:	Optimization_Tool		
Issue Links:	Dependency		
	depends on CRD-17 The Optimization Simulator shall calc...		Quality Check passed
	depends on CRD-18 The Optimization Simulator shall pass...		Quality Check passed
Requirement Type:	Functional		
Event / Use Case:	UC-6.2, UC-6.3, UC-6.4, UC-6.7, UC-6.8		
Rationale:	Fitness function defines the goal of swarm behaviour		
Fit Criterion:	Optimization tool is able to rank the candidate controllers according fitness score		
Customer Satisfaction:	very high		
Customer Dissatisfaction:	very high		
Source:	Modelling-Optimization Requirements Workshop Bonn (8-9-17)		
History:	Reformulated RE-75		
Supporting Material:	D3.1		

[CRD-21] The Modelling tool should be able to present the structural diagram of a swarm member											
Status:	Quality Check passed										
Project:	CPSwarm - Requirements and Development										
Type:	Volere Requirement	Priority:	Major								
Reporter:	Sarah Suleri	Assignee:	Alessandra Bagnato								
Labels:	Modelling_Tool										
Issue Links:	<table border="1"> <thead> <tr> <th colspan="4">Dependency</th> </tr> </thead> <tbody> <tr> <td>is a dependency of</td> <td>CRD-30</td> <td>The modelling tool shall enable users...</td> <td>Quality Check passed</td> </tr> </tbody> </table>			Dependency				is a dependency of	CRD-30	The modelling tool shall enable users...	Quality Check passed
Dependency											
is a dependency of	CRD-30	The modelling tool shall enable users...	Quality Check passed								
Requirement Type:	Functional										
Rationale:	To provide an overview structure of a swarm member										
Fit Criterion:	The user should see the components and connections										
Customer Satisfaction:	very high										
Customer Dissatisfaction:	high										
Source:	Requirement follow-up telco										
Supporting Material:	D5.2, D2.3										

[CRD-22] Modelling library shall include a library to help in designing a swarm member			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Medium
Reporter:	Sarah Suleri	Assignee:	Alessandra Bagnato
Labels:	Library, Modelling_Tool		
Issue Links:	Dependency		
	depends on	CRD-1 The Modelling library will be a collec...	Quality Check passed
	is a dependency of	CRD-34 Swarm member library shall contain mo...	Quality Check passed
	is a dependency of	CRD-25 The swarm member library shall contain...	Quality Check passed
	is a dependency of	CRD-26 The swarm member library shall contain...	Quality Check passed
	is a dependency of	CRD-27 The swarm member library shall contain...	Quality Check passed
Requirement Type:	Functional		
Rationale:	Re-usability of pre-existing models of swarm members		
Fit Criterion:	The swarm member library should contain enough components to put together a minimum viable swarm member		
Customer Satisfaction:	high		
Customer Dissatisfaction:	high		
Source:	Requirement follow-up telco		
Supporting Material:	D4.1		

[CRD-23] The Modelling library shall include a library to help in designing an environment															
Status:	Quality Check passed														
Project:	CPSwarm - Requirements and Development														
Type:	Volere Requirement	Priority:	Medium												
Reporter:	Sarah Suleri	Assignee:	Alessandra Bagnato												
Labels:	Library, Modelling_Tool														
Issue Links:	<table border="1"> <thead> <tr> <th colspan="4">Dependency</th> </tr> </thead> <tbody> <tr> <td>depends on</td> <td>CRD-1</td> <td>The Modelling library will be a collec...</td> <td>Quality Check passed</td> </tr> <tr> <td>is a dependency of</td> <td>CRD-28</td> <td>The environment library shall contain...</td> <td>Quality Check passed</td> </tr> </tbody> </table>			Dependency				depends on	CRD-1	The Modelling library will be a collec...	Quality Check passed	is a dependency of	CRD-28	The environment library shall contain...	Quality Check passed
Dependency															
depends on	CRD-1	The Modelling library will be a collec...	Quality Check passed												
is a dependency of	CRD-28	The environment library shall contain...	Quality Check passed												
Requirement Type:	Functional														
Rationale:	Re-usability of pre-existing models of environment														
Fit Criterion:	The environment library should have enough components for modelling basic environment for simulation														
Customer Satisfaction:	high														
Customer Dissatisfaction:	high														
Source:	Requirement follow-up telco														
Supporting Material:	D4.1														

[CRD-24] The Modelling library shall include a library to help in designing a goal															
Status:	Quality Check passed														
Project:	CPSwarm - Requirements and Development														
Type:	Volere Requirement	Priority:	Medium												
Reporter:	Sarah Suleri	Assignee:	Alessandra Bagnato												
Labels:	Library, Modelling_Tool														
Issue Links:	<table border="1"> <thead> <tr> <th colspan="4">Dependency</th> </tr> </thead> <tbody> <tr> <td>depends on</td> <td>CRD-1</td> <td>The Modelling library will be a collec...</td> <td>Quality Check passed</td> </tr> <tr> <td>is a dependency of</td> <td>CRD-29</td> <td>The goal library shall contain variou...</td> <td>Quality Check passed</td> </tr> </tbody> </table>			Dependency				depends on	CRD-1	The Modelling library will be a collec...	Quality Check passed	is a dependency of	CRD-29	The goal library shall contain variou...	Quality Check passed
Dependency															
depends on	CRD-1	The Modelling library will be a collec...	Quality Check passed												
is a dependency of	CRD-29	The goal library shall contain variou...	Quality Check passed												
Requirement Type:	Functional														
Rationale:	Re-usability of pre-existing goals														
Fit Criterion:	The goals library should include common mathematically defined fitness functions														
Customer Satisfaction:	high														
Customer Dissatisfaction:	high														
Source:	Requirement follow-up telco														
Supporting Material:	D4.1														

[CRD-25] The swarm member library shall contain models for the physical aspects of the swarm member												
Status:	Quality Check passed											
Project:	CPSwarm - Requirements and Development											
Type:	Volere Requirement	Priority:	Major									
Reporter:	Sarah Suleri	Assignee:	Alessandra Bagnato									
Labels:	Library, Modelling_Tool											
Issue Links:	<table border="1"> <thead> <tr> <th colspan="3">Dependency</th> </tr> </thead> <tbody> <tr> <td>depends on CRD-1</td> <td>The Modelling library will be a collec...</td> <td>Quality Check passed</td> </tr> <tr> <td>depends on CRD-22</td> <td>Modelling library shall include a libr...</td> <td>Quality Check passed</td> </tr> </tbody> </table>			Dependency			depends on CRD-1	The Modelling library will be a collec...	Quality Check passed	depends on CRD-22	Modelling library shall include a libr...	Quality Check passed
Dependency												
depends on CRD-1	The Modelling library will be a collec...	Quality Check passed										
depends on CRD-22	Modelling library shall include a libr...	Quality Check passed										
Requirement Type:	Functional											
Rationale:	Re-usability of pre-existing sensors and actuators of swarm member											
Fit Criterion:	It should contain enough components to put together a minimum viable swarm member											
Customer Satisfaction:	high											
Customer Dissatisfaction:	high											
Source:	Requirement follow-up telco											
Supporting Material:	D4.1											

[CRD-26] The swarm member library shall contain models for the behaviour of a swarm member												
Status:	Quality Check passed											
Project:	CPSwarm - Requirements and Development											
Type:	Volere Requirement	Priority:	Major									
Reporter:	Sarah Suleri	Assignee:	Alessandra Bagnato									
Labels:	Library, Modelling_Tool											
Issue Links:	<table border="1"> <thead> <tr> <th colspan="3">Dependency</th> </tr> </thead> <tbody> <tr> <td>depends on CRD-1</td> <td>The Modelling library will be a collec...</td> <td>Quality Check passed</td> </tr> <tr> <td>depends on CRD-22</td> <td>Modelling library shall include a libr...</td> <td>Quality Check passed</td> </tr> </tbody> </table>			Dependency			depends on CRD-1	The Modelling library will be a collec...	Quality Check passed	depends on CRD-22	Modelling library shall include a libr...	Quality Check passed
Dependency												
depends on CRD-1	The Modelling library will be a collec...	Quality Check passed										
depends on CRD-22	Modelling library shall include a libr...	Quality Check passed										
Requirement Type:	Functional											
Rationale:	Re-usability of pre-existing behaviour model of swarm member											
Fit Criterion:	It should contain enough components to put together a minimum viable swarm member											
Customer Satisfaction:	high											
Customer Dissatisfaction:	high											
Source:	Requirement follow-up telco											
Supporting Material:	D4.1											

[CRD-27] The swarm member library shall contain models for the communication among swarm members												
Status:	Quality Check passed											
Project:	CPSwarm - Requirements and Development											
Type:	Volere Requirement	Priority:	Major									
Reporter:	Sarah Suleri	Assignee:	Alessandra Bagnato									
Labels:	Library, Modelling_Tool											
Issue Links:	<table border="1"> <thead> <tr> <th colspan="3">Dependency</th> </tr> </thead> <tbody> <tr> <td>depends on CRD-1</td> <td>The Modelling library will be a collec...</td> <td>Quality Check passed</td> </tr> <tr> <td>depends on CRD-22</td> <td>Modelling library shall include a libr...</td> <td>Quality Check passed</td> </tr> </tbody> </table>			Dependency			depends on CRD-1	The Modelling library will be a collec...	Quality Check passed	depends on CRD-22	Modelling library shall include a libr...	Quality Check passed
Dependency												
depends on CRD-1	The Modelling library will be a collec...	Quality Check passed										
depends on CRD-22	Modelling library shall include a libr...	Quality Check passed										
Requirement Type:	Functional											
Rationale:	Re-usability of pre-existing communication models of swarm											
Fit Criterion:	It should contain enough components to put together a minimum viable swarm member											
Customer Satisfaction:	high											
Customer Dissatisfaction:	high											
Source:	Requirement follow-up telco											
Supporting Material:	D4.1											

[CRD-28] The environment library shall contain models of environments			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Minor
Reporter:	Farshid Tavakolizadeh	Assignee:	Alessandra Bagnato
Labels:	Modelling_Tool		
Issue Links:	<p style="text-align: center;">Dependency</p> <p>depends on CRD-23 The Modelling library shall include a ... Quality Check passed</p>		
Requirement Type:	Functional		
Fit Criterion:	The library has at least one model for each vision scenario Each model should contain size (width/height), resolution, and a map of the environment. The map should contain paths and obstacles.		
Customer Satisfaction:	high		
Customer Dissatisfaction:	neutral		
Source:	Modelling & optimization requirements telco		
Supporting Material:	D4.2		

[CRD-29] The goal library shall contain various fitness functions linked to different problems			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Minor
Reporter:	Farshid Tavakolizadeh	Assignee:	Alessandra Bagnato
Labels:	Modelling_Tool		
Issue Links:	<p style="text-align: center;">Dependency</p> <p>depends on CRD-24 The Modelling library shall include a... Quality Check passed</p>		
Requirement Type:	Functional		
Rationale:	To reuse predefined fitness functions		
Fit Criterion:	The library has enough fitness functions to realise the vision scenarios		
Customer Satisfaction:	high		
Customer Dissatisfaction:	neutral		
Source:	Modelling & optimisation requirements telco		
Supporting Material:	D4.2		

[CRD-30] The modelling tool shall enable users to create models and publish them in a private library			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Trivial
Reporter:	Sarah Suleri	Assignee:	Etienne Brosse
Labels:	Modelling_Tool		
Issue Links:	<p style="text-align: center;">Dependency</p> <p>depends on CRD-21 The Modelling tool should be able to p... Quality Check passed</p>		
Requirement Type:	Functional		
Rationale:	To reuse and manage private models		
Fit Criterion:	You can export and import models to a private library		
Customer Satisfaction:	neutral		
Customer Dissatisfaction:	neutral		
Source:	Modelling and Optimization Telco		
Supporting Material:	D5.2, D5.4, D5.3		

[CRD-31] The modelling tool shall contain an editor to formulate the fitness function			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Sarah Suleri	Assignee:	Etienne Brosse
Labels:	Modelling_Tool		
Issue Links:	<div style="background-color: #f4d03f; padding: 5px; text-align: center;">Dependency</div> depends on CRD-7 The Modelling tool shall be able to mo... Quality Check passed		
Requirement Type:	Functional		
Rationale:	To enable to the user to formulate a fitness function regarding a specific problem		
Fit Criterion:	At least be able to specify basic mathematical expression		
Customer Satisfaction:	very high		
Customer Dissatisfaction:	high		
Source:	Modelling & optimization telco		
Supporting Material:	D5.3,D5.4		

[CRD-32] The modelling tool shall be able to model the behaviour of the swarm member using the swarm member behaviour library			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Critical
Reporter:	Sarah Suleri	Assignee:	Etienne Brosse
Labels:	Library, Modelling_Tool		
Issue Links:	<div style="background-color: #f4d03f; padding: 5px; text-align: center;">Dependency</div> depends on CRD-3 The Modelling tool shall be able to mo... Quality Check passed		
Requirement Type:	Functional		
Rationale:	To define the behaviour of the swarm member. To define what a swarm member can do		
Fit Criterion:	The model should contain at least one behaviour from the library		
Customer Satisfaction:	very high		
Customer Dissatisfaction:	very high		
Source:	Modelling & Optimization telco		
Supporting Material:	D4.1		

[CRD-33] Modelling tool shall be able to model a local state as a part of the swarm member structure			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Critical
Reporter:	Sarah Suleri	Assignee:	Etienne Brosse
Labels:	Modelling_Tool		
Issue Links:	Bonfire Testing		
	Testing discovered	CRD-36 The modelling tool shall provide the ...	Quality Check passed
Issue Links:	Dependency		
	depends on	CRD-3 The Modelling tool shall be able to mo...	Quality Check passed
Requirement Type:	Functional		
Rationale:	The swarm member can inform about its local status		
Fit Criterion:	Behaviour can store and retrieve the local state of the swarm member		
Customer Satisfaction:	very high		
Customer Dissatisfaction:	very high		
Source:	Modelling & Optimization telco		
Supporting Material:	D4.1		

CRD-34] Swarm member library shall contain models for sensors and actuators to be used to design a swarm member			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Sarah Suleri	Assignee:	Etienne Brosse
Labels:	Modelling_Library		
Issue Links:	<div style="background-color: #f4d03f; padding: 5px; text-align: center;">Dependency</div> depends on CRD-22 Modelling library shall include a libr... Quality Check passed		
Requirement Type:	Functional		
Rationale:	Reusability of sensors and actuators		
Fit Criterion:	The library should contain a minimum viable number of sensors and actuators to design a swarm member		
Customer Satisfaction:	high		
Customer Dissatisfaction:	high		
Source:	Requirement engineering telco		
Supporting Material:	D4.1		

[CRD-35] The communication link between the swarm and the Monitoring Tool shall be authenticated and encrypted			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Critical
Reporter:	Sarah Suleri	Assignee:	René Reiners
Labels:	Monitoring_tool, Security		
Requirement Type:	Functional		
Rationale:	Data received from swarm needs to stay confidential. The confidential data received from the swarm should not be accessed by unauthorized entity. Protection of the swarm from attacks.		
Fit Criterion:	The encryption and authentication should be SOA		
Customer Satisfaction:	very high		
Customer Dissatisfaction:	very high		
Source:	RE Workshop Bonn 15.11.17		
History:	Reformulated Issue #100		
Supporting Material:	GitLab Issue #100 D4.7, D4.8 D7.5, D7.6		

[CRD-36] The modelling tool shall provide the type of swarm member, type of data and data source to the monitoring tool											
Status:	Quality Check passed										
Project:	CPSwarm - Requirements and Development										
Type:	Volere Requirement	Priority:	Major								
Reporter:	Artiza Elosegui	Assignee:	Alessandra Bagnato								
Labels:	Modelling_Tool										
Issue Links:	<table border="1"> <thead> <tr> <th colspan="4">Bonfire Testing</th> </tr> </thead> <tbody> <tr> <td>Discovered testing</td> <td>while CRD-33</td> <td>Modelling tool shall be able to model...</td> <td>Quality Check passed</td> </tr> </tbody> </table>			Bonfire Testing				Discovered testing	while CRD-33	Modelling tool shall be able to model...	Quality Check passed
Bonfire Testing											
Discovered testing	while CRD-33	Modelling tool shall be able to model...	Quality Check passed								
Requirement Type:	Functional										
Fit Criterion:	<ul style="list-style-type: none"> • Type of the swarm member (Some number or string) - What type of agent am I talking to • Type of Data (struct, uint8...) – What data are we transferring number, structures, arrays • Data Source (/drone/pos) – Where do we get the data from (topics, variables, etc) 										
Customer Satisfaction:	Very high										
Customer Dissatisfaction:	Very high										

[CRD-37] The configuration tool shall provide the type and address of swarm member			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Artiza Elosegui	Assignee:	René Reiners
Labels:	Monitoring_tool		
Requirement Type:	Functional		
Rationale:	To enable the swarm operator to monitor the swarm		
Fit Criterion:	<ul style="list-style-type: none"> • Type of Swarm member <ul style="list-style-type: none"> o Type of Data – type of the element to be monitored o Data Source – location of where to get the element to be monitored • Address of the Swarm member(IP/ID) - How to contact the agent 		
Customer Satisfaction:	high		
Customer Dissatisfaction:	high		
Source:	TTTech		

[CRD-54] The modelling tool shall be responsible for passing swarm member structure to the code generator			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Sarah Suleri	Assignee:	Alessandra Bagnato
Labels:	Code_generator, Modelling_Tool		
Requirement Type:	Functional		
Rationale:	This info is needed for generating driver specific code		
Fit Criterion:	Code generator understands the swarm member structure passed		
Customer Satisfaction:	high		
Customer Dissatisfaction:	high		
Source:	RE Follow up Telco		
Supporting Material:	D3.1		

[CRD-55] [The modelling tool shall be responsible for passing swarm member behaviour to the code generator](#)

Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Sarah Suleri	Assignee:	Alessandra Bagnato
Labels:	Modelling_Tool		
Requirement Type:	Functional		
Rationale:	This info is needed for generating controller specific code		
Fit Criterion:	Code generator understands the swarm member behaviour passed		
Customer Satisfaction:	high		
Customer Dissatisfaction:	high		
Source:	RE Follow up Telco		
Supporting Material:	D3.1, D5.2		

[CRD-56] The optimization tool shall pass the optimal behaviour to the code generator			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Sarah Suleri	Assignee:	Micha Rappaport
Labels:	Optimization_Tool		
Requirement Type:	Functional		
Rationale:	This info is needed for generating controller specific code		
Fit Criterion:	Code generator understands the swarm member behaviour passed		
Customer Satisfaction:	high		
Customer Dissatisfaction:	high		
Source:	RE Follow up Telco		
Supporting Material:	D3.1, D5.2		

[CRD-57] The Code Generator shall generate code for a specific target platform			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Critical
Reporter:	Sarah Suleri	Assignee:	Gianluca Prato
Labels:	Code_generator		
Requirement Type:	Functional		
Rationale:	This is needed for deployment/execution on target devices		
Fit Criterion:	Generated code can be deployed and executed at the target platform		
Customer Satisfaction:	high		
Customer Dissatisfaction:	high		
Source:	RE Follow-up Telco		
Supporting Material:	D3.1, D7.2, D7.1		

[CRD-58] The Deployment Tool shall deploy artefacts on swarm members			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Minor
Reporter:	Farshid Tavakolizadeh	Assignee:	Farshid Tavakolizadeh
Labels:	Deployment_Tool		
Requirement Type:	Functional		
Issue Links:	Extension		
	is extended by	CRD-59 The Deployment Agent shall report the...	Quality Check passed
	is extended by	CRD-78 The Deployment Agent shall use the li...	Quality Check passed
	is extended by	CRD-79 The Deployment Agent shall be respons...	Quality Check passed
	Related		
	is related to	CRD-60 The communication between the Deploy...	Quality Check passed
is related to	CRD-61 The Deployment Manager shall receive ...	Quality Check passed	
Rationale:	This is needed to enable mass deployment on remote devices (without physical access, without exposed interfaces)		
Fit Criterion:	The artefacts can be deployed to remote devices in bulks		
Customer Satisfaction:	very high		
Customer Dissatisfaction:	high		
Source:	D7.2.1 D7.2.2		
Supporting Material:	This is needed to enable mass deployment on remote devices (without physical access, without exposed interfaces)		

Description

The generated code shall be either:

- executable on the target platform
- raw code with instructions on how to be compiled on target

[CRD-59] The Deployment Agent shall report the deployment status			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Farshid Tavakolizadeh	Assignee:	Farshid Tavakolizadeh
Labels:	Deployment_Tool		
Issue Links:	Extension		
	extends	CRD-58 The Deployment Tool shall deploy arte...	Quality Check passed
	is extended by	CRD-60 The communication between the Deploy...	Quality Check passed
	Related		
	is related to	CRD-61 The Deployment Manager shall receive ...	Quality Check passed
Requirement Type:	Functional		
Rationale:	Functional		
Fit Criterion:	The status of deployment is required in order to monitor and synchronise software updates (automatically or by operators)		
Customer Satisfaction:	The status of bulk deployment tasks are reported in a format understandable by machines and humans		
Customer Dissatisfaction:	high		
Source:	very high		
Supporting Material:	D7.2.1 D7.2.2		

Description

The deployment status contains information about the state of the deployment, reasons for failure, and possibly log messages. Deployment Agent shall offer the possibility of reporting this information back to the Deployment Manager.

[CRD-60] The communication between the Deployment Agent running on swarm members and the Deployment Manager shall be authenticated, authorized, encrypted, and integrity checked.			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Farshid Tavakolizadeh	Assignee:	Farshid Tavakolizadeh
Labels:	Deployment_Tool		
Issue Links:	Extension		
	extends	CRD-59	The Deployment Agent shall report the... Quality Check passed
	Inclusion		
	is included by	CRD-73	The Deployment Tool shall implement s... Quality Check passed
	Meta Issue		
is part of	CRD-67	All communications between the swarm ... Quality Check passed	
Related			
is related to	CRD-58	The Deployment Tool shall deploy arte... Quality Check passed	
Requirement Type:	Functional		
Rationale:	Secure deployments are vital for secure operation of swarms.		
Fit Criterion:	All security aspects including authentication, authorisation, encryption, and package signature validation are taken into account during deployment tasks.		
Customer Satisfaction:	high		
Customer Dissatisfaction:	high		
Source:	D7.2.1 D7.2.2		
Supporting Material:	Functional		

Description

- Data transmitted to and received from swarm needs to stay confidential.
- Only authorised entities should be able to transmit data to the swarm members.
- The confidential data received from the swarm should not be accessed by unauthorized entities.
- Data received from the deployment server must be validated.
- The privilege of deploying software artefacts on swarm members is separate from the generic privilege of being a member of the swarm.

[CRD-61] The Deployment Manager shall receive the configuration of the deployment task from the operator prior to deployment			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Farshid Tavakolizadeh	Assignee:	Farshid Tavakolizadeh
Labels:	Deployment_Tool		
Issue Links:	Related		
	is related to CRD-59 The Deployment Agent shall report the... Quality Check passed		
	is related to CRD-58 The Deployment Tool shall deploy arte... Quality Check passed		
Requirement Type:	Functional		
Rationale:	Deployment tool requires the configuration of the deployment task to know how and where to deploy artefacts.		
Fit Criterion:	Deployment Tool can be used to target specific or a group of swarm members to deploy different types of artefacts		
Customer Satisfaction:	high		
Customer Dissatisfaction:	high		
Source:	D7.2.1 D7.2.2		
Supporting Material:	Deployment tool requires the configuration of the deployment task to know how and where to deploy artefacts.		

Description

Deployment tool requires the configuration of the deployment which is a procedure on how (required steps) and where (target swarm members) to deploy artefacts.

[CRD-62] The Modelling Tool shall make it possible to define events.			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Regina Bíró	Assignee:	Etienne Brosse
Labels:	Modelling_Tool, Security		
Issue Links:	Dependency		
	is a dependency of	CRD-65 The Modelling Tool shall distinguish ...	Quality Check passed
	is a dependency of	CRD-66 The Modelling Tool shall make it poss...	Quality Check passed
	is a dependency of	CRD-69 The Modelling Tool shall make it poss...	Quality Check passed
	is a dependency of	CRD-77 The Modelling Tool shall make it poss...	Quality Check passed
	is a dependency of	CRD-82 The Modelling Library shall include s...	Quality Check passed
	is a dependency of	CRD-84 The Modelling Library shall include b...	Quality Check passed
	is a dependency of	CRD-86 The Modelling Library shall include a...	Quality Check passed
	is a dependency of	CRD-93 The Monitoring and Configuration Tool...	Quality Check passed
	is a dependency of	CRD-74 Components in the Modelling Library c...	Quality Check passed
Requirement Type:	Functional		
Rationale:	In order to model event-driven behavior changes for safety and security reasons, the user has to be able to define events in the Modelling Tool.		
Fit Criterion:	The Modelling Tool can be used to model event-driven behavior switches		
Customer Satisfaction:	very high		

Customer Dissatisfaction:	very high
Source:	
Supporting Material:	

[CRD-63] The Code Generator shall generate code that is readable and understandable by humans.			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Balint Janvari	Assignee:	Gianluca Prato
Labels:	Code_generator, Security		
Requirement Type:	Functional		
Rationale:	Generated code may need to be extended by hand or reviewed and audited at later stages of the project.		
Fit Criterion:	Names must be descriptive, formatting must be consistent and all code sections should be annotated with comments to describe functionality.		
Customer Satisfaction:	high		
Customer Dissatisfaction:	high		
Source:			
Supporting Material:			

Description

Generated code should follow clear, secure-by-default coding conventions. The code must be human readable, annotated with comments as necessary, and should follow a consistent, descriptive naming convention.

[CRD-64] The Code Generator and all the code generated shall be compliant to ISO 26262.			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Balint Janvari	Assignee:	Gianluca Prato
Labels:	Code_generator, Security		
Requirement Type:	Functional		
Rationale:	Ensuring that the code generator outputs code that is compliant with best practices described in the standard ensures that it can be used in automotive use cases.		
Fit Criterion:	Refer to requirements as defined in the standard.		
Customer Satisfaction:	very high		
Customer Dissatisfaction:	neutral		
Source:			
Supporting Material:			

Description

ISO 26262 / IEC 61508 is a standard for the design of safety critical systems.

[CRD-65] The Modelling Tool shall distinguish between swarm, member and component scope events, which are defined at their respective level in the model hierarchy.			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Regina Bíró	Assignee:	Etienne Brosse
Labels:	Modelling_Tool, Security		
Issue Links:	Dependency		
	depends on	CRD-62 The Modelling Tool shall make it poss...	Quality Check passed
	is a dependency of	CRD-66 The Modelling Tool shall make it poss...	Quality Check passed
	is a dependency of	CRD-69 The Modelling Tool shall make it poss...	Quality Check passed
	is a dependency of	CRD-77 The Modelling Tool shall make it poss...	Quality Check passed
	is a dependency of	CRD-82 The Modelling Library shall include s...	Quality Check passed
	is a dependency of	CRD-84 The Modelling Library shall include b...	Quality Check passed
	is a dependency of	CRD-86 The Modelling Library shall include a...	Quality Check passed
	is a dependency of	CRD-93 The Monitoring and Configuration Tool...	Quality Check passed
	is a dependency of	CRD-100 The Modelling Tool shall make it poss...	Quality Check passed
Requirement Type:	Functional		
Rationale:	Swarm scope events are propagated to other members of the swarm, while member scope events are not. Component scope events are like member scope events, but cannot be triggered remotely and are imported from the Modelling Library alongside their respective component.		
Fit Criterion:	The Modelling Tool can be used to model event-driven behavior switches, where		

	events have different labels based on their occurrences.
Customer Satisfaction:	very high
Customer Dissatisfaction:	very high
Source:	
Supporting Material:	

[CRD-66] The Modelling Tool shall make it possible to trigger events based on the current value of the inputs and outputs defined for the low-level behavior of the current state.				
Status:	Quality Check passed			
Project:	CPSwarm - Requirements and Development			
Type:	Volere Requirement	Priority:	Major	
Reporter:	Regina Bíró	Assignee:	Etienne Brosse	
Labels:	Modelling_Tool, Security			
Issue Links:	Dependency			
	depends on	CRD-62	The Modelling Tool shall make it poss...	Quality Check passed
	depends on	CRD-65	The Modelling Tool shall distinguish ...	Quality Check passed
	is a dependency of	CRD-69	The Modelling Tool shall make it poss...	Quality Check passed
	is a dependency of	CRD-77	The Modelling Tool shall make it poss...	Quality Check passed
	is a dependency of	CRD-82	The Modelling Library shall include s...	Quality Check passed
	is a dependency of	CRD-84	The Modelling Library shall include b...	Quality Check passed
	is a dependency of	CRD-86	The Modelling Library shall include a...	Quality Check passed
	is a dependency of	CRD-93	The Monitoring and Configuration Tool...	Quality Check passed
Requirement Type:	Functional			
Rationale:	To be able to model event-driven behavior, we need to specify which kind of inputs can induce events and thus behavior changes.			
Fit Criterion:	Events can be triggered based on the value of sensor, actuator and behavioral inputs and outputs.			
Customer Satisfaction:	very high			
Customer Dissatisfaction:	very high			

Source:	
Supporting Material:	

[CRD-67] All communications between the swarm and the tools in the workbench shall be authenticated, integrity protected and encrypted.			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Balint Janvari	Assignee:	Balint Janvari
Labels:	Deployment_Tool, Monitoring_tool, Security		
Issue Links:	Meta Issue		
	contains CRD-60	The communication between the Deploym...	Quality Check passed
Requirement Type:	Functional		
Rationale:	Updating software, setting parameters and issuing commands are sensitive operation by their very nature.		
Fit Criterion:	All communications between the swarm and the tools in the workbench must use industry standard encryption and signature schemes.		
Customer Satisfaction:	high		
Customer Dissatisfaction:	high		
Source:			
Supporting Material:			

Description

Deployment and monitoring should only be possible after authentication and with proper authorization. Messages in transit should be treated as confidential and must be protected against tampering and eavesdropping.

[CRD-68] All communications between swarm members shall be authenticated and integrity protected, with a per-message policy on encryption.			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Balint Janvari	Assignee:	Balint Janvari
Labels:	Security		
Requirement Type:	Functional		
Rationale:	Outside parties should not be able to fabricate events.		
Fit Criterion:	All communications between swarm members should use industry standard signature schemes.		
Customer Satisfaction:	high		
Customer Dissatisfaction:	high		
Source:			
Supporting Material:			

Descriptions

Communications between swarm members mostly include remote events. Certain safety critical events need no encryption, and in fact benefit from reduced latency.

[CRD-69] The Modelling Tool shall make it possible to add additional swarm scope events to each state transition that are triggered when the transition happens.			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Regina Bíró	Assignee:	Etienne Brosse
Labels:	Modelling_Tool, Security		
Issue Links:	Dependency		
	depends on	CRD-62	The Modelling Tool shall make it poss... Quality Check passed
	depends on	CRD-65	The Modelling Tool shall distinguish ... Quality Check passed
	depends on	CRD-66	The Modelling Tool shall make it poss... Quality Check passed
	is a dependency of	CRD-80	The Modelling Tool shall allow compon... Quality Check passed
	is a dependency of	CRD-77	The Modelling Tool shall make it poss... Quality Check passed
	is a dependency of	CRD-82	The Modelling Library shall include s... Quality Check passed
	is a dependency of	CRD-84	The Modelling Library shall include b... Quality Check passed
	is a dependency of	CRD-86	The Modelling Library shall include a... Quality Check passed
	is a dependency of	CRD-93	The Monitoring and Configuration Tool... Quality Check passed
Requirement Type:	Functional		
Rationale:	To make it possible to e.g. trigger a behavior change event which is generated remotely by another swarm member or is generated by the operator and is valid for the whole swarm		

Fit Criterion:	Events can be triggered by other events.
Customer Satisfaction:	neutral
Customer Dissatisfaction:	high
Source:	
Supporting Material:	

[CRD-70] The physical interface used for all communications and its parameters shall be configurable using the Modelling Tool.			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Balint Janvari	Assignee:	Etienne Brosse
Labels:	Code_generator, Modelling_Tool, Security		
Requirement Type:	Functional		
Rationale:	Parameters such as per message encryption, physical layer and so on are part of the basic design of the system, and as such, should be configurable from the Modelling Tool.		
Fit Criterion:	The Modelling Tool supports communication interface models and the Code Generator uses that information to target the correct interface.		
Customer Satisfaction:	high		
Customer Dissatisfaction:	high		
Source:			
Supporting Material:			

Description

The communication facilities used should be modelled alongside the rest of the CPS.

[CRD-71] [The Modelling Library shall include CPS components for communication interfaces, with support for at least one type of mesh network.](#)

Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Balint Janvari	Assignee:	Etienne Brosse
Labels:	Modelling_Library, Modelling_Tool, Security		
Requirement Type:	Functional		
Rationale:	Enhances user experience and ties in lower level functionality.		
Fit Criterion:	The count of such models should be at least 1, with at least 1 of them being models for interfaces connecting to mesh networks.		
Customer Satisfaction:	high		
Customer Dissatisfaction:	high		
Source:			
Supporting Material:			

Description

These built-in models enhance user experience by allowing common interfaces to be inserted into a project with very little effort.

[CRD-72] The Deployment Manager shall sign all packages with an operator specific key.			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Balint Janvari	Assignee:	Farshid Tavakolizadeh
Labels:	Deployment_Tool, Security		
Issue Links:	Inclusion		
	is included by	CRD-73 The Deployment Tool shall implement s...	Quality Check passed
Issue Links:	Mention		
	mentions	CRD-75 The Deployment Agent shall verify the...	Quality Check passed
Requirement Type:	Functional		
Rationale:	In order to maintain strict control over the main binary, it should only ever be started or stopped by the Deployment Agent. Before updates, it would need to be stopped anyways, and it makes signature validations before startups a lot simpler.		
Fit Criterion:	The main binary should only ever be started by the designated instance of the Deployment Agent.		
Customer Satisfaction:	high		
Customer Dissatisfaction:	high		
Source:			
Supporting Material:			

Description

The Deployment Agent should take upon itself the burden of managing the life-cycle of the main binary.

[CRD-81] <u>Software components running on the CPS shall be started with the lowest possible privileges.</u>			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Balint Janvari	Assignee:	Balint Janvari
Labels:	Security		
Requirement Type:	Functional		
Rationale:	Reduces attack surface and isolates critical system components.		
Fit Criterion:	None of the project related processes running on the CPS have unnecessary privileges. All necessary privileges are explained and documented.		
Customer Satisfaction:	neutral		
Customer Dissatisfaction:	high		
Source:			
Supporting Material:			

Description

On a Unix-like system, this might mean user level separation, containerization, capability dropping and namespacing. In essence, even if one of our components is compromised, it should not be able to do much harm.

[CRD- 82] The Modelling Library shall include special behaviors to support the built-in behaviors of the CPS like emergency shutdown, flying home or refueling.			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Regina Bíró	Assignee:	Etienne Brosse
Labels:	None		
Issue Links:	Dependency		
	depends on	CRD-62 The Modelling Tool shall make it poss...	Quality Check passed
	depends on	CRD-65 The Modelling Tool shall distinguish ...	Quality Check passed
	depends on	CRD-66 The Modelling Tool shall make it poss...	Quality Check passed
	depends on	CRD-69 The Modelling Tool shall make it poss...	Quality Check passed
	depends on	CRD-77 The Modelling Tool shall make it poss...	Quality Check passed
	is a dependency of	CRD-84 The Modelling Library shall include b...	Quality Check passed
	is a dependency of	CRD-86 The Modelling Library shall include a...	Quality Check passed
	is a dependency of	CRD-89 The Monitoring and Configuration Tool...	Quality Check passed
is a dependency of	CRD-92 The Monitoring and Configuration Tool...	Quality Check passed	
Requirement Type:	Functional		
Rationale:	While these behaviors are often implemented as lower-level solutions, in some cases the user would want to define them as contingency behaviors which can be triggered by certain events.		
Fit Criterion:	The above described behaviors are part of the Modelling Library.		
Customer	very high		

Satisfaction:	
Customer Dissatisfaction:	high
Source:	
Supporting Material:	

[CRD-83] [The Abstraction Layer shall have low level support for remote shutdown requests that works regardless the status of the current behavior.](#)

Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Balint Janvari	Assignee:	Gianluca Prato
Labels:	Abstraction_Layer, Security		
Requirement Type:	Functional		
Rationale:	Shutdown requests in emergency situations should work regardless of bugs in the behavior or its current state.		
Fit Criterion:	Certain types of shutdown requests are processed by the Abstraction Layer at such a low level that even if the main behavior is in an infinite loop, it could come into effect within a reasonable amount of time.		
Customer Satisfaction:	high		
Customer Dissatisfaction:	high		
Source:			
Supporting Material:			

Description

Remote shutdown request should have a way to interrupt the main behavior.

[CRD-84] The Modelling Library shall include behaviors specific to target hardware platforms that can be used as safe default contingency plans for each CPS model (soft shutdown).

Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Regina Bíró	Assignee:	Etienne Brosse
Labels:	Modelling_Library, Security		
Issue Links:	Dependency		
	depends on	CRD-62	The Modelling Tool shall make it poss... Quality Check passed
	depends on	CRD-65	The Modelling Tool shall distinguish ... Quality Check passed
	depends on	CRD-66	The Modelling Tool shall make it poss... Quality Check passed
	depends on	CRD-69	The Modelling Tool shall make it poss... Quality Check passed
	depends on	CRD-77	The Modelling Tool shall make it poss... Quality Check passed
	depends on	CRD-82	The Modelling Library shall include s... Quality Check passed
	depends on	CRD-74	Components in the Modelling Library c... Quality Check passed
	is a dependency of	CRD-86	The Modelling Library shall include a... Quality Check passed
is a dependency of	CRD-89	The Monitoring and Configuration Tool... Quality Check passed	
Requirement Type:	Functional		
Rationale:	Safe default contingency plans could include to retreat to a safe location with pre-given GPS coordinates or any other more complex behavior.		
Fit Criterion:	Examples for more complex safe contingency behaviors are part of the Modelling Library.		
Customer	very high		

Satisfaction:	
Customer Dissatisfaction:	high
Source:	
Supporting Material:	

[CRD-85] [The Abstraction Layer shall implement a hardware specific safe remote shutdown behavior that cannot be overridden by the current behavior \(hard shutdown\).](#)

Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Balint Janvari	Assignee:	Gianluca Prato
Labels:	Abstraction_Layer, Security		
Requirement Type:	Functional		
Rationale:	Hard shutdowns are device and not goal specific.		
Fit Criterion:	For each supported hardware platform, the Abstraction Library has such a built-in shutdown behavior.		
Customer Satisfaction:	high		
Customer Dissatisfaction:	high		
Source:			
Supporting Material:			

Description

Normal shutdown requests are processed by the main behavior to support graceful, soft shutdowns. Hard shutdown requests, on the other hand, should trigger a built-in behavior in the CPS that is specific to the target platform and performs a safe shutdown as soon as possible.

[CRD-86] The Modelling Library shall include a special behavior that switches over the CPS to manual remote control.			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Regina Bíró	Assignee:	Etienne Brosse
Labels:	Modelling_Library, Security		
Issue Links:	Dependency		
	depends on	CRD-62	The Modelling Tool shall make it poss... Quality Check passed
	depends on	CRD-65	The Modelling Tool shall distinguish ... Quality Check passed
	depends on	CRD-66	The Modelling Tool shall make it poss... Quality Check passed
	depends on	CRD-69	The Modelling Tool shall make it poss... Quality Check passed
	depends on	CRD-77	The Modelling Tool shall make it poss... Quality Check passed
	depends on	CRD-82	The Modelling Library shall include s... Quality Check passed
	depends on	CRD-84	The Modelling Library shall include b... Quality Check passed
	depends on	CRD-74	Components in the Modelling Library c... Quality Check passed
	is a dependency of	CRD-89	The Monitoring and Configuration Tool... Quality Check passed
Requirement Type:	Functional		
Rationale:	In emergency situations the operator might want to take over a swarm member. This should be modeled as a behavior to which a swarm member can switch to after receiving the signal for remote control takeover.		
Fit Criterion:	The switch to remote control behavior is part of the Modelling Library.		
Customer	very high		

Satisfaction:	
Customer Dissatisfaction:	high
Source:	
Supporting Material:	

[CRD-87] [The Modelling Tool shall let multiple high-level behaviors coexist within the same project.](#)

Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Balint Janvari	Assignee:	Etienne Brosse
Labels:	Modelling_Tool, Security		
Requirement Type:	Functional		
Rationale:	Helps design and simulate complex project with heterogeneous behavior.		
Fit Criterion:	More than one high level behavior can exist in a project.		
Customer Satisfaction:	high		
Customer Dissatisfaction:	high		
Source:			
Supporting Material:			

Description

To allow swarms and swarm simulations with heterogeneous behavior, the Modelling Tool should support multiple different behaviors in the same project. For each behavior, code generation should be performed separately.

[CRD-88] <u>The Simulation Manager shall support simulations where different swarm members have different behaviors.</u>			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Balint Janvari	Assignee:	Davide Conzon (ISMB)
Labels:	Security, Simulation_Tool		
Requirement Type:	Functional		
Rationale:	Required for both optimization and simple simulation, in order to simulate heterogeneous swarms.		
Fit Criterion:	Simulations can be launched where some swarm members have a different behavior than other swarm members.		
Customer Satisfaction:	high		
Customer Dissatisfaction:	high		
Source:			
Supporting Material:			

Description

Some interface needs to exist to assemble to configuration of the swarm.

[CRD-89] The Monitoring and Configuration Tool shall be able to trigger remote events on individual swarm members.			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Regina Bíró	Assignee:	Artiza Elosegui
Labels:	Monitoring_tool, Security		
Issue Links:	Dependency		
	depends on CRD-82 The Modelling Library shall include s...	Quality Check passed	
	depends on CRD-84 The Modelling Library shall include b...	Quality Check passed	
	depends on CRD-86 The Modelling Library shall include a...	Quality Check passed	
Requirement Type:	Functional		
Rationale:	The Swarm Operator might want to give remote orders to swarm members during operation.		
Fit Criterion:	Swarm members respond to the triggered events accordingly.		
Customer Satisfaction:	very high		
Customer Dissatisfaction:	very high		
Source:			
Supporting Material:			

[CRD-91] The Optimization Tool shall only optimize one behavior at a time, but shall let the simulation used include other behaviors .			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Balint Janvari	Assignee:	Wilfried Elmenreich
Labels:	Optimization_Tool, Security		
Requirement Type:	Functional		
Rationale:	To optimize behavior in heterogeneous swarms and in the presence of malicious actors.		
Fit Criterion:	Optimizations can be run with arbitrary swarm configurations.		
Customer Satisfaction:	high		
Customer Dissatisfaction:	high		
Source:			
Supporting Material:			

[CRD-92] The Monitoring and Configuration Tool shall enable the user to launch an external tool to take remote control of a specific swarm member.			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Regina Bíró	Assignee:	Artiza Elosegui
Labels:	Monitoring_tool, Security		
Issue Links:	<p style="text-align: center;">Dependency</p> <p>depends on CRD-82 The Modelling Library shall include s... Quality Check passed</p>		
Requirement Type:	Functional		
Rationale:	The Swarm Operator might want to take over a swarm member in case of an emergency.		
Fit Criterion:	It is possible to control the swarm member via the tool launched.		
Customer Satisfaction:	very high		
Customer Dissatisfaction:	very high		
Source:			
Supporting Material:			

[CRD-99] Abstraction Layer shall provide primitives to activate and control high-level CPS routines.			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Gianluca Prato	Assignee:	Gianluca Prato
Labels:	Abstraction_Layer		
Requirement Type:	Functional		
Rationale:	In order to define complex behaviours a set of routines should be provided as base functionalities		
Fit Criterion:	A set of higher level functionalities that do not need to be optimized using the Optimization Tool or do not imply swarm behaviour will be provided by the Abstraction Layer.		
Customer Satisfaction:	high		
Customer Dissatisfaction:	very high		
Source:			
Supporting Material:			

Description

High level routines could be used as starting block to compose a behaviour state-machine.

[CRD-100] The Modelling Tool shall make it possible to specify event scope.			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Regina Bíró	Assignee:	Etienne Brosse
Labels:	Modelling_Tool, Security		
Issue Links:	<div style="background-color: #f4b084; text-align: center; padding: 5px;">Dependency</div> depends on CRD-65 The Modelling Tool shall distinguish ... Quality Check passed		
Requirement Type:	Functional		
Rationale:			
Fit Criterion:	If scoped as a swarm event, events are for the entire swarm, while member scope events are events specific to a given swarm member.		
Customer Satisfaction:	high		
Customer Dissatisfaction:	neutral		
Source:			
Supporting Material:			

Description

The modelling tool is able to specify the scope of modelled events.

Which means that swarm scope events are events for the entire swarm, while member scope events are internal events i.e. internal to a specific swarm member.

[CRD-101] The Modelling Tool shall namespace component scope events to their respective component			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Medium
Reporter:	Regina Bíró	Assignee:	Etienne Brosse
Labels:	None		
Issue Links:	<div style="background-color: #f4d03f; padding: 5px; text-align: center;">Dependency</div> depends on CRD-74 Components in the Modelling Library c... Quality Check passed		
Requirement Type:	Functional		
Rationale:			
Fit Criterion:	Scope events are distinguished even when multiple components exist of the same type.		
Customer Satisfaction:	high		
Customer Dissatisfaction:	neutral		
Source:			
Supporting Material:			

Description

If multiple components of the same type exist on the same CPS, their respective component scope events need to be distinguished.

[CRD-102] The Code Generator shall integrate low-level behavior algorithms implemented manually.			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Regina Bíró	Assignee:	Gianluca Prato
Labels:	Code_generator		
Requirement Type:	Functional		
Rationale:	Some behaviors not generated by the Optimization Tool but are handwritten in form of code need to be integrated during the behaviour-code generation process.		
Fit Criterion:	The Code generator can generate code integrating custom algorithms written by hand.		
Customer Satisfaction:	high		
Customer Dissatisfaction:	high		
Source:	Security Workshop, Budapest		
Supporting Material:			

[CRD-103] The Deployment Tool shall provide the means to compile codes on target platforms			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Minor
Reporter:	Farshid Tavakolizadeh	Assignee:	Angel Soriano
Labels:	Deployment_Tool		
Issue Links:	Inclusion		
	is included by	CRD-105 The Deployment Tool shall provide the...	Quality Check passed
	Related		
	is related to	CRD-104 The Deployment Tool shall provide the...	Quality Check passed
Requirement Type:	Functional		
Rationale:	Native compilation is less complex when dealing with different hardware and software architectures on robotic systems.		
Fit Criterion:	Deployment Tool offers the possibility of native compilation on target devices.		
Customer Satisfaction:	high		
Customer Dissatisfaction:	low		
Source:	<p>Emails [2018.04.19-2018.04.26]</p> <p>1. Do you always cross-compile your codes and then deploy or you always compile on the target? [ROB] We usually compile on the target because not every robots have the same hardware and software architecture and it is always easier to avoid any problem with dependences, etc.. [TTT] Currently we are compiling on the platform itself.</p> <p>[DGSKY] We usually cross-compile the codes and then deploy on the target.</p> <p>2. Can you claim that either of (cross-compilation/compilation on target) is always the best approach? (consider time, dependencies, linking, complexity, etc.) [ROB] For us, for our daily work, the compilation on the target is the best approach because we usually work with the same laptop but with different robots. But in this case maybe it can be different because the idea is get the same code for several robots with the same hardware & software. Therefore, it might be interesting to obtain a</p>		

	<p>cross-compilation method on an external laptop/server that, if the compilation is satisfactory, automatically sends the code to each robot. This is something that we have never done before.</p> <p>[TTT] We cannot claim that compilation on the target is always the best approach. Neither that cross-compilation is the best approach.</p> <p>[DGSKY] We consider cross-compilation our best approach because in the most of case we work with the same target hardware (companion computer) and we can get advantage of powerfull host computer for compilation and then an easy deployment on targets. Cross-compilation requires a good toolchain setup that in some case means some hours of (hard) work but after that we gain time during compilation and deployment.</p>
Supporting Material:	

Description

When compilation is required, the Deployment Tool should be able to move generated codes to target devices and compile them using provided build scripts. The build script may setup or rely on pre-existing build dependencies on the target build environment.

[CRD-104] The Deployment Tool shall provide the means to cross-compile codes for the target platforms			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Farshid Tavakolizadeh	Assignee:	Omar Morando
Labels:	Deployment_Tool		
Issue Links:	Inclusion		
	is included by	CRD-105 The Deployment Tool shall provide the...	Quality Check passed
	Related		
	is related to	CRD-103 The Deployment Tool shall provide the...	Quality Check passed
Requirement Type:	Functional		
Rationale:	Cross-compilation benefits from powerful host machines and saves time when targeting similar hardware/software platforms.		
Fit Criterion:	Deployment Tool offers the possibility of cross-compilation for target platforms.		
Customer Satisfaction:	high		
Customer Dissatisfaction:	high		
Source:	<p>Emails [2018.04.19-2018.04.26]</p> <p>1. Do you always cross-compile your codes and then deploy or you always compile on the target? [ROB] We usually compile on the target because not every robots have the same hardware and software architecture and it is always easier to avoid any problem with dependences, etc.. [TTT] Currently we are compiling on the platform itself.</p> <p>[DGSKY] We usually cross-compile the codes and then deploy on the target.</p> <p>2. Can you claim that either of (cross-compilation/compilation on target) is always the best approach? (consider time, dependencies, linking, complexity, etc.) [ROB] For us, for our daily work, the compilation on the target is the best approach because we usually work with the same laptop but with different robots. But in this case maybe it can be different because the idea is get the same code for several robots</p>		

	<p>with the same hardware & software. Therefore, it might be interesting to obtain a cross-compilation method on an external laptop/server that, if the compilation is satisfactory, automatically sends the code to each robot. This is something that we have never done before.</p> <p>[TTT] We cannot claim that compilation on the target is always the best approach. Neither that cross-compilation is the best approach.</p> <p>[DGSKY] We consider cross-compilation our best approach because in the most of case we work with the same target hardware (companion computer) and we can get advantage of powerfull host computer for compilation and then an easy deployment on targets. Cross-compilation requires a good toolchain setup that in some case means some hours of (hard) work but after that we gain time during compilation and deployment.</p>
Supporting Material:	

Description

When compilation is required, the Deployment Tool should be able to execute build scripts that cross-compile source codes locally, before sending and installing them on the targets.

[CRD-105] The Deployment Tool shall provide the means to compile codes			
Status:	Quality Check passed		
Project:	CPSwarm - Requirements and Development		
Type:	Volere Requirement	Priority:	Major
Reporter:	Farshid Tavakolizadeh	Assignee:	Artiza Elosegui
Labels:	Deployment_Tool		
Issue Links:	Inclusion		
	includes CRD-104 The Deployment Tool shall provide the...		Quality Check passed
	includes CRD-103 The Deployment Tool shall provide the...		Quality Check passed
Requirement Type:	Functional		
Rationale:	Code compilation is required when codes in compiled programming languages are being deployed.		
Fit Criterion:	Deployment Tool is able to compile codes using provided build scripts		
Customer Satisfaction:	high		
Customer Dissatisfaction:	high		
Source:			
Supporting Material:			

Description

When compilation is required, the Deployment Tool should be able to execute build scripts that compile codes for/on target platforms. The tool shall support cross-compilation ([CRD-104](#)) at first and then be extended to support native compilation () on target devices.

4.3 Requirements Validation

Validation activities done by SLAB are divided throughout the project lifetime into two tasks: T2.4 Validation Framework Specification and T8.4 Use Cases Validation. The deliverable D2.8 describing the Validation Framework (produced by T2.4) due M18 will include a methodology established by SLAB which will be used to validate the requirements created for the CPSwarm Workbench. Here we describe the main characteristics and workflow of the Validation Framework designed for the CPSwarm project.

Validation and Verification are procedures in quality management checking whether a product, service or system meets its predefined requirements and whether it fulfils its intended purpose. Since these two terms are often used together, sometimes interchangeably, it is worth taking some words to clarify what we mean by validation and verification in the CPSwarm project. Validation aims to answer the question *"Are we building the right system?"* whereas Verification helps us to answer to *"Are we building the system right?"* Validation is used to ensure that a product, service or system is designed to satisfy the needs of its customers, users and other stakeholders while verification ensures that the end product complies with its specification.

Our methodology uses different kind of metrics to validate and verify requirements – namely Key Performance Indicators (KPIs), Test Cases and Maturity Levels.

First, the requirements are translated into measurable metrics: either test cases which, when passed indicate that the requirement has been met, or into KPIs which set a target value in a way that supports the assumption that the requirement has been met. Templates for KPIs and formal/informal test cases are included in the deliverable.

When a requirement meets the KPI assigned to it or passes its test case, it indicates that the project is making progress – but to measure how much, these events are linked to specific maturity levels. Thus, different KPIs are required for different maturity levels. We defined five maturity levels to be used in the Validation Framework:

1. Proof of concept (demonstrates feasibility)
2. Working (core features are present)
3. Feature complete (all planned features are present)
4. Optimized (performance is up to expectations, reasonably error free)
5. Production ready (meets standards, has documentation, easy to use)

Building on the roadmap described in the project proposal, we set a number of milestones based on the due dates of relevant deliverables with target maturity levels for each component and the workbench as a whole.

The goal of our validation activities is to track and validate changes to the project requirements – implementing an iterative approach. When requirements change and/or components mature, these changes are periodically registered and new metrics are tailored to validate them.

5 Lessons Learned

The following sections discuss lessons learned for various components of the CPSwarm workbench. We have used a standard template [18] in order to document lessons learned.

5.1 Modelling Library

Table 3: Lessons learned for Modelling Library

Category	Issue Name	Problem/Success	Impact	Recommendation
Lifecycle	Modelling Library required updates	Softeam is in charge of updating the Modelling Library. This includes new example of modelling component.	Users are expecting to find their models inside the modelling library as soon as they made them.	Meeting or teleconference would be useful to decide which component should be part of the Modelling Library.
Wording	Confusion between Modelling Library and Modelling Catalogue	Both Modelling Library and Modelling Catalogue concepts have been introduced in CPSwarm. Modelling Library is composed of reusable modelling component as Modelling Catalogue is composed of Model example.	Modeller want to publish their model sometime is unclear if it should be part of the Modelling library or Modelling Catalogue.	Try to clarify when it is Modelling Library or Modelling Catalogue.

5.2 Modelling

Table 4: Lessons learned for Modelling

Category	Issue Name	Problem/Success	Impact	Recommendation
Modelling	Dynamic environment	Dynamic environment seems to be needed, but an environment can be dynamic in many ways and it will depend of the problem but also Simulator capability	Hard to define what is a dynamic environment, what can or must be modelled specially if it is not used by simulator	The concept of dynamic environment must be refined according to targeted simulator and their capabilities.
Interface	Behaviour format	Modelling tool must model Swarm Member behaviour and pass it to the Optimization Tool, Code generation Tool and Simulator.	Modelling tool have to be able to transform its internal representation of CPS Member behaviour (state machine) into one or many format which must be relevant for all other component.	The choice of the chosen format for representing the Swarm member behavioural should be, if possible, the same for all component and conform to some standard.
Interface	Swarm composition format	Modelling tool must model Swarm member composition and pass it to the Optimization Tool, and Simulator.	Modelling tool have to be able to transform its internal representation of Swarm Member composition into one or many format which must be relevant for all other component.	The choice of the chosen format for representing the Swarm member composition should be, if possible, the same for all component and conform to some standard.
Interface	Environment format	Modelling tool must model	Modelling tool have to be	The choice of the chosen simulator

Category	Issue Name	Problem/Success	Impact	Recommendation
		Swarm Environment to the Simulator.	able to transform its internal representation of Swarm Environment into one or many format relevant for all targeted simulator.	should be limited in order to limit the number of possible export from the Environment simulation.

5.3 Optimization

Table 5: Lessons learned for Optimization

Category	Issue Name	Problem/Success	Impact	Recommendation
Interface	Optimization Tool API	Interface passes problem description as Java source code file from Modelio to FREVO.	FREVO requires recompilation after modelling in Modelio.	Create a generic problem component and only pass parameters as XML file to it.
Interface	Optimization Tool API	Interface passes problem description as Java source code file from Modelio to FREVO.	Existing swarm algorithms are not implemented as problem descriptions, but as representations (in FREVO) and are not evolved in FREVO.	Add a second part to the java file transmitted from Modelio and FREVO for swarm algorithms in form of representations.
Functionality	Human-in-the-loop	Evolving human-in-the-loop parameters during optimization of fitness function.	The parameters (modelled in the Modelling Tool) should be optimized with the fitness function. This implies an increased computation time for the evolutionary process.	The parameters are part of the fitness function. Furthermore, we should propose a measure, related to processing time (how many parameters are reasonable to be evolved)
Component	Optimization Tool	FREVO is now the only instance for the optimization tool.	We do not know anything about the recommendations of other	Research: other optimization tools – what are their requirements (especially for

Category	Issue Name	Problem/Success	Impact	Recommendation
			optimization tools.	interfaces), what is their focus/expected output

5.4 Simulation

Table 6: Lessons learned for Simulation

Category	Issue Name	Problem/Success	Impact	Recommendation
Component Integration	Integrating simulation environment	A simple Java simulation environment has been integrated with the broker based approach.	The code is specific for the current simulator type.	Develop the code to integrate the simulation environment as general as possible, to make it simple to be reused with other simulators.
Component	Optimization Simulator	The optimization simulator cannot be configured just, by passing parameters.	Simulation in optimization simulator needs to be implemented manually.	Include certain simulations in the modelling library.
Interface	Simulator API	With the broker based simulator API, the agent controller resides within FREVO.	The message passing between FREVO and the simulator consumes a lot of time. This makes the optimization process expressively long.	Pass the agent controller candidate to the optimization simulator.
Interface	Simulator API	FREVO needs to discover simulation servers during optimization.	The discovery process congests communication at the broker and inhibits scalability with more than two simulation servers.	Use a dedicated discovery phase and cache simulation server addresses to establish direct connections.
Component integration	Integrating simulation environment	Using a broker to control remotely heterogeneous simulation	The Stage simulator has been started using	A simple stage simulation can be started using the current approach,

Category	Issue Name	Problem/Success	Impact	Recommendation
		environments	messages received through an MQTT broker.	but to control a complete Stage simulation, there will be the need to pass more information.

5.5 Code Generation

This task is started only at M13. For this reason, a set of lessons learned is not yet available. Otherwise, it is possible to provide a list of recommendation for the design and development of the code generator.

Table 7: Lessons learned for Code Generation

Category	Issue Name	Problem/Success	Impact	Recommendation
Code generator input	NA	NA	NA	Code generator should receive a description of the CPS (sensors, actuators, functionalities).
Code generator input	NA	NA	NA	Code generator should receive a description (in some standardized format) of the CPS behaviour.
Code generator input	NA	NA	NA	Code generator should receive information related to the software target platform, e.g. ROS, Python, and C++ (internally we named it as "Runtime Environment").
Code generator output	NA	NA	NA	The output of the code generator should be the complete source code of the behaviour of a CPS, and not the corresponding HW-specific machine code.

5.6 Abstraction Layer

Table 8: Lessons learned for Abstraction Layer

Category	Issue Name	Problem/Success	Impact	Recommendation
Testing Robotic Hardware Platform	Selection of best candidate Abstraction Library from SotA.	Testing ROS (Robot Operating System) on a simple platform that not support it natively (AlphaBot-Pi 2).	ROS has been easily installed on the platform.	Abstraction Layer should be not implemented from scratch but part of the implementation could take advantage of what is already provided by ROS operating system.

Testing ROS	Support of multiple programming languages.	Testing compatibility between ROS nodes developed with different programming languages. (C++/Python)	Integration of C++ ROS nodes with Python ROS nodes is feasible.	Developing the Abstraction Library starting from ROS could take advantage of the easy integration of different programming languages (C++/Python).
-------------	--	--	---	--

5.7 Deployment

Table 9: Lessons learned for Deployment

Category	Issue Name	Problem/Success	Impact	Recommendation
Deployment	Message delivery behind firewalls	Deployment of software onto devices behind firewalls.	Deployment using traditional approaches (e.g. via SSH) will be difficult	Use a publish/subscribe communication protocols to deliver instructions and artefacts to devices (subscribers)
Monitoring	Deployment monitoring	Deployment of software must be monitored to detect failures	Lack of monitoring mechanism leaves the system at risk during and after deployment	Perform necessary tests and transmit the results after every update
Deployment	Versioning	Multiple swarm members may need compatible software versions to inter-operate	Mismatching software versions will make the system unusable, possibly breaking the swarm behaviour	Allow version control for every deployment step in order to keep devices in sync and roll back during failures

5.8 Monitoring

Table 10: Lessons learned for Monitoring

Category	Issue Name	Problem/Success	Impact	Recommendation
Communication Library	Selection of most suitable	A survey on the most suitable	NA	Use a widely used protocol such as

Selection	communication library	communication library is ongoing		MQTT or Some/IP
-----------	-----------------------	----------------------------------	--	-----------------

5.9 Continuous Integration

Table 11: Lessons learned for Continuous Integration

Category	Issue Name	Problem/Success	Impact	Recommendation
Testing Environment	Testing Modelling Tool	Testing GUI application in a command-line environment is problematic.	The graphical elements cannot be executed.	Using a virtual GUI makes it possible to test elements of GUI application in the command-line CI environment.
Testing Environment	Testing Optimization Tool	Frevo requires a running simulator during certain tests.	Simultaneous tests may create unwanted influence on each other.	Run an isolated instance of simulator for every test session.

6 Design Pattern Library

Throughout the requirements elicitation process in the first 16 months of the projects, it turned out that there is more than “just” functional and non-functional requirements and “user needs” that need to be taken into account when designing a cyber-physical system in a specific domain. There is a need of a certain knowledge foundation on which the team can start designing applications. This knowledge foundation includes knowledge, e.g., on:

- Rules and regulations
- Safety rules and law
- Proven design guidelines
- Standards
- De-facto standards
- Commonly known agreements and workarounds
- UI Design

This set of categories is not complete but will be further exploited throughout the second half of the project. Knowledge from these areas is usually not directly formulated as a requirement but something that needs to be taken in to account when trying to fulfil other requirements. Thus, this knowledge is implicit and only accessible via domain experts. The approach of the CPSwarm project is to gather technologies and knowledge in libraries as described in the preceding sections. For the kind of knowledge described here, the common knowledge ground cannot be kept, explained and conveyed in a technical way.

Therefore, the concept of *Application Design Patterns* is applied in this concept and interconnected pieces of knowledge will be formulated within an evolving design pattern library. The concept of the design pattern is established since long in architecture [19], software systems [20], organizational contexts [21] as well as user-interface [22], website [23] or application design [24].

A design patterns consists of specific parts describing the *context*, in which a certain *problem* occurs and how it can be *solved*, including the *consequences* of the solution. Patterns are usually organized in clusters of design pattern libraries that are presented by domain experts after a long time of engineering. CPSwarm follows the approach of described in [25] who formulates patterns during system engineering and design time. Pattern mature and reach a reliable state during the project work and are supported by the project community instead of one pattern expert. The whole approach is web-based such that the CPSwarm project can present and make accessible the patterns gathered over the project duration and beyond. This helps to present sustainable domain knowledge supporting future work.

In upcoming deliverables from work packages 4 that handle the “Human-in-the-Loop” concept, the implemented approach will be described in more detail together with the online version of the CPSwarm Design Pattern Library.

7 Conclusion

The initial phase of the CPSwarm project focused on the specification of use cases, the definition of its stakeholders, as well as the description of the communication flow between them. Beyond, it 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 analysis presented in this deliverable started with the description of the Volere requirements scheme that is used throughout this deliverable to specify requirements. The process of requirement engineering in D2.6 proceeds by taking one step forward from the user needs and requirements identified in D2.3. In D2.3 we extracted user roles who interact with workbench and alter the communication flow between them by dividing it into four phases; Design, Implementation, Deployment and Operation phase. From the perspective of each user role, we defined user needs in the form of user stories. The next step was to translate these user needs into abstract workbench components and to define flow of information between them. The responsibilities of these workbench components and the data flow between them were defined in the form of technical requirements. D2.6 contains the reiteration of these user needs and requirements. In addition to the requirements, D2.6 also contains an initial set of lessons learned during the design and implementation of various components of the workbench.

The requirements specified in this version of the deliverable are to be seen as an intermediate 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 user needs and Volere requirements, this deliverable D2.6 laid the foundation that will be used in further implementation in the technical WPs.

Conclusively, this deliverable documented the iterative process of ideation and concept development in order to identify various user needs. In addition to the user needs, the identification and specification of Volere requirements related to the workbench components and their lessons learned are significant results from this task that will be used as input to subsequent activities of the project.

Acronyms

Acronym	Explanation
CPS	Cyber Physical System
CI	Continuous Integration
GUI	Graphical User Interface
HW	Hardware

List of figures

Figure 1: The CPSwarm Workbench development lifecycle	6
Figure 2: The Volere "Requirements Shell" for representing atomic requirements [11]	7
Figure 3: Screenshot of the user need creation dialog in CPSwarm	8
Figure 4: Screenshot of the Volere Requirement creation dialog in CPSwarm	9
Figure 5: State diagram of issue type "User Need"	10
Figure 6: State diagram of issue type "Volere Requirement"	10

List of tables

Table 1: User needs.....	11
Table 2: Workbench component descriptions.....	12
Table 3: Lessons learned for Modelling Library	101
Table 4: Lessons learned for Modelling.....	102
Table 5: Lessons learned for Optimization.....	103
Table 6: Lessons learned for Simulation.....	104
Table 7: Lessons learned for Code Generation.....	105
Table 8: Lessons learned for Abstraction Layer.....	105
Table 9: Lessons learned for Deployment.....	106
Table 10: Lessons learned for Monitoring.....	106
Table 11: Lessons learned for Continuous Integration.....	108

References

- [1] Chin, G., M.B. Rosson, and J.M. Carroll. Participatory analysis: shared development of requirements from scenarios. In SIGCHI conference on Human factors in computing systems. 1997.
- [2] Easterbrook, S., Negotiation and the Role of the Requirements Specification. Appears in P. Quintas (ed.) Social Dimensions of Systems Engineering: People, processes, policies and software development, 1993: p. 144-164.
- [3] Glinz, M., Improving the Quality of Requirements with Scenarios, in Proceeding of the Second World Conference on Requirements Engineering. 2000: Schaumburg. p. 254-271.
- [4] Holbrook H., I., A scenario-based methodology for conducting requirements elicitation SIGSOFT Softw. Eng. Notes 1990 15 (1): p. 95-104

- [5] ISO, ISO 9241-210:2010 Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems. International Organization for Standardization, 2010.
- [6] Jarke, M. and K. Pohl, Requirements engineering in 2001: (virtually) managing a changing reality. IEEE Software Engineering, 1994. 9(6).
- [7] Penna, G.D., et al., An XML Definition Language to Support Scenario-Based Requirements Engineering. International Journal of Software Engineering and Knowledge Engineering, 2003. 13(3): p. 237-256.
- [8] Ramesh, B. and M. Jarke, Toward Reference Models for Requirements Traceability. IEEE Trans. Softw. Eng., 2001. 27(1): p. 58-93.
- [9] Robertson, J. and Robertson, S.; Mastering the Requirements Process. 1999: Addison-Wesley.
- [10] Robertson, J. and Robertson, S.; Requirements-Led Project Management. 2004: Addison-Wesley.
- [11] Robertson, J. and Robertson, S.; Volere Requirements Specification Template. 2010.
- [12] Stufflebeam, W., A.I. Antón, and T.A. Alspaugh. SMaRT – Scenario Management and Requirements Tool. In 11th IEEE International Requirements Engineering Conference. 2003.
- [13] Sutcliffe, A.G., W.-C. Chang, and R. Neville, Evolutionary Requirements Analysis, in 11th IEEE Requirements Engineering Conference. 2003.
- [14] Sutcliffe, A.G. Scenario-based Requirements Engineering. In 11th IEEE International Requirements Engineering Conference (RE'03). 2003.
- [15] Zimmerman, J., Stolterman, E., and Forlizzi, J. An Analysis and Critique of Research through Design: Towards a Formalization of a Research Approach. In Proceedings of the 8th ACM Conference on Designing Interactive Systems, 2010: ACM Press, p. 310–319.
- [16] Design, Venture, The Inter-disciplinarian, and About Me. "Your Best Agile User Story". Alex Cowan. N.p., 2017. Web. 18 June 2017.
- [17] Gomaa, H. and D.B.H. Scott. Prototyping as a tool in the specification of user requirements. In Proceedings of the 5th international conference on Software engineering. 1981. San Diego, California, United States.
- [18] M. Piscopo, "Lessons Learned," Lessons Learned Template. [Online]. Available: <http://www.projectmanagementdocs.com/project-closing-templates/lessons-learned.html#axzz55ll5w9dT>. [Accessed: 04-Jan-2018]
- [19] Alexander, C.; A Pattern Language: Towns, Buildings, Construction. Oxford University Press, New York, NY, USA, 1977.
- [20] Gamma, E. and Helm, R. and Johnson, R. and Vlissides, J.; Design Patterns. Elements of Reusable Object-Oriented Software. Addison-Wesley Longman, Amsterdam, 1994.
- [21] Manns, M., and Rising L.; Fearless Change: Patterns for Introducing New Ideas: Introducing Patterns into Organizations. Addison-Wesley, Boston, MA, USA, 2005.
- [22] Tidwell J.; Designing Interfaces. O'Reilly Media, Sebastopol, CA, USA, 2nd edition, 2011.
- [23] van Duyne, D., and Landay J., and Hong J.; The Design of Sites: Patterns for Creating Winning Websites. Prentice Hall, 2nd edition, 2007.
- [24] Borchers, J.; A Pattern Approach to Interaction Design. John Wiley & Sons, West Sussex, England, 1st edition, 2001.
- [25] Reiners, R.; An Evolving Pattern Library for Collaborative Project Documentation. Shaker, 2014.