DELIVERABLE

D6.1 – Design Technology Background

Project Title	**COMP4DRONES**
Grant Agreement number	826610
Call and topic identifier	H2020-ECSEL-2018
Funding Scheme	Research & Innovation Action (RIA)
Project duration	36 Months [1 October 2019 – 30 September 2022]
Coordinator	Mr. Rodrigo Castiñeira (INDRA)
Website	www.**COMP4DRONES**.eu

ECSEL Joint Undertaking
Electronic Components and Systems for European Leadership

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826610

Document fiche			
Authors:	Partner No	Partner Name	Contributors
	2.	**AIT**	Christoph Schmittner and Rupert Schlick
	6.	**BUT**	Pavel Zemcik, Peter Chudy and Svetozar Nosko
	7.	**UWB**	Miroslav Flídr and Ondřej Severa
	10.	**ENAC**	Gautier Hattenberger
	11.	**SIEMENS**	Federico Cappuzzo
	13.	**ENSMA**	Yassine Ouhammou
	15.	**UNIMORE**	Alessandro Capotondi and Andrea Marongiu
	16.	**UNISANNIO**	Giuseppe Silano, Valerio Mariani and Luigi Iannelli
	17.	**UNISS**	Francesca Palumbo, Luca Pulina and Tiziana Fanni
	18.	**UNIVAQ**	Vittoriano Muttillo
	23.	**ANYWI**	Morten Larsen and Jens Finkhaeuser
	30.	**ACORDE**	Fernando Herrera and David Abia
	33.	**IKERLAN**	Liher Granado and Lorea Belategi
	34.	**CEA**	Ansgar Radermacher
	35.	**UNICAN**	Eugenio Villar
	37.	**SM**	Petr Barták
	44.	**SHERPA**	Muhammad Tufail and Fabrice Peyrin
	52.	**ALM**	Ludo Stellingwerff
	53.	**ALTRAN**	Vincent Bompart and Guillaume Thalmann
Internal reviewers:	Elodie Renault [SCALIAN] Federico Corradi [IMEC-NL],		
Work Package:	WP6		
Task:	T6.1-3		
Nature:	R		
Dissemination:	PU		

Document History			
Version	**Date**	**Contributor(s)**	**Description**
V0.1	20-5-2020	Eugenio Villar	Initial template to be filled by partners
V0.17	17-6-2020	WP6 partners	Contributions by partners
V0.18	23-6-2020	Eugenio Villar	First draft for WP6 partner's review
V0.19	26-6-2020	Eugenio Villar	Final draft for WP6 partner's review
V1.0	03-7-2020	Eugenio Villar	Final draft for review
V1.1	15-7-2020	Eugenio Villar	Final version

Keywords:	Design Tools, V-Cycle
Abstract (few lines):	This Deliverable describes the design methods and tools brought by the technology partners as background technology

DISCLAIMER

ACKNOWLEDGEMENT

This document is a deliverable of **COMP4DRONES** project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nº 826610

Table of Contents

Table of Figures

Definitions, Acronyms and Abbreviations

Acronym	Title
AR/VR	Augmented Reality/Virtual Reality
C4D-DTF	C4D Design Tool Framework
ConOps	Concept of Operations
CGR	Coarse-Grained Reconfigurable
CPSoS	Cyber-Physical Systems of Systems
COTS	Commercial Off the Shelf
DOF	Degrees of Freedom
DP	DronePort
DTC	Design Technology
FPGA	Field Programmable Gate Array
GNSS	Global Navigation Satellite System
HAL	Hardware Abstraction Layer
HWPE	Hardware Processing Element
IIoT	Industrial Internet of Things
IPS	Indoor Positioning system
MDC	Multi-Dataflow Composer
MiL	Model in the Loop
NN	Neural Network
OMPL	Open Motion Library
RISC-V	RISC-V ISA Processing Unit
SoC	System on Chip
SoS	Systems of Systems
SIL	Software-in-the-loop
UAV	Unmanned Aerial Vehicle
UC	Use Case
UGV	Unmanned Ground Vehicle

Executive Summary

In this deliverable, the background design technologies provided by the partners are described. They refer to design methods and tools able to improve the process of designing, verifying, and deploying software and hardware drone architectures enabling cost-effective compositional design and assurance of drone modules and systems.

Their description, specific installation and/or functional requirements and the improvements to be made during the course of C4D will be described.

1 C4D System Engineering Framework

1.1 Introduction

As described in Deliverable D2.1, Design Tools are one of the Key Enabling Technologies that are needed to have a fully functioning drone system with the required quality, cost, performance and design time. The different tools developed will be integrated in an Engineering Framework and Development Workbench adapted to drone applications, the C4D Design Framework. The framework, eventually with additional, currently available commercial and/or open-source tools, will cover both sides of the V-Cycle for HW/SW CPSoS as shown in Figure 1.

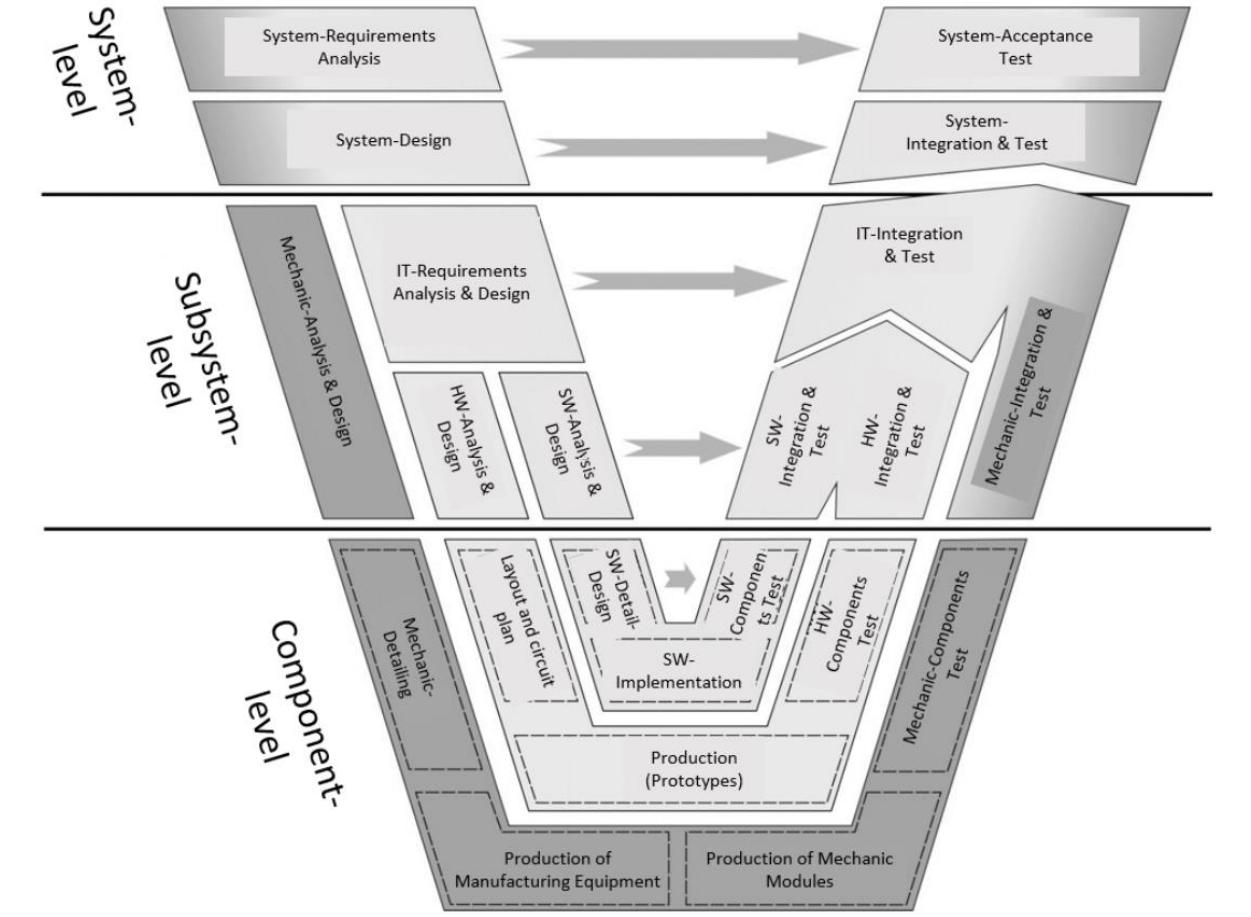


Figure 1 V-Cycle for CPSoS.

The V-Cycle above extends the traditional SW Engineering V-cycle to mechatronic systems. In C4D this part of the system is essential as the drone has to be addressed as a product in which HW and SW electronics including sensors is integral to the mechanics of the drone, the motors, the battery and the physical aero-dynamics involved in its flying.

C4D WP6 does not pretend to build a single engineering tool by integrating contributions from the partners. The objective is to identify the weaknesses in current drone modeling, validation, design and verification tools so that they can be improved.

The current tooling landscape for engineering drone-embedded systems is quite fragmented. It is characterized by a series of isolated tools with different and sometimes complementary capabilities ranging from specialized coding towards sophisticated simulation environments. Moreover, current

modeling and design practices and tools still need harmonization to enable agility and interoperability and clear separation of concerns for efficient disaggregation of engineering roles at different abstraction levels. Additionally, as other domains leading with services provided as complex CPSoS, currently available tools lack of the required levels of scalability and reusability.

Additionally, as **COMP4DRONES** follows a modular and incremental approach with specific computation and communication mechanisms (e.g., partitioning, protocols, and contracts), tooling must be capable to be customized to manage configurations in a composable and agile way.

The result will be the C4D Engineering Framework and Development Workbench where different tools will inter-operate with already available commercial tools facilitating the modeling verification and design of complex services based on drones.

1.2 The C4D System Engineering Framework

Design tools were grouped in D2.1 in two groups: tools for service specification and tools for system development, as shown in 0 In each group, main tasks were identified.

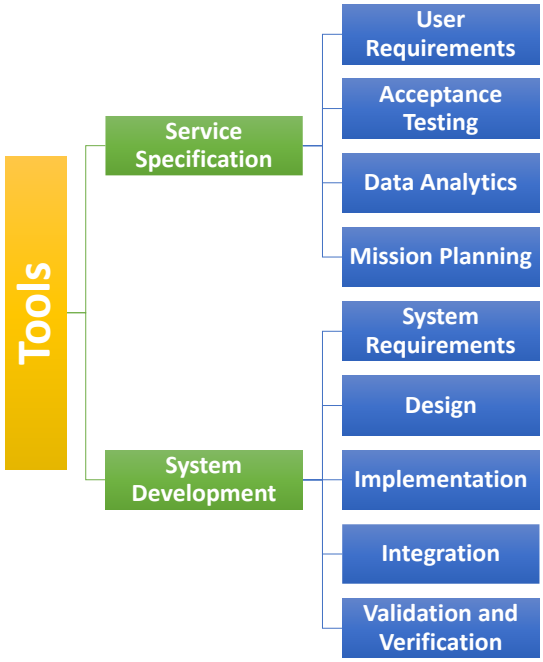
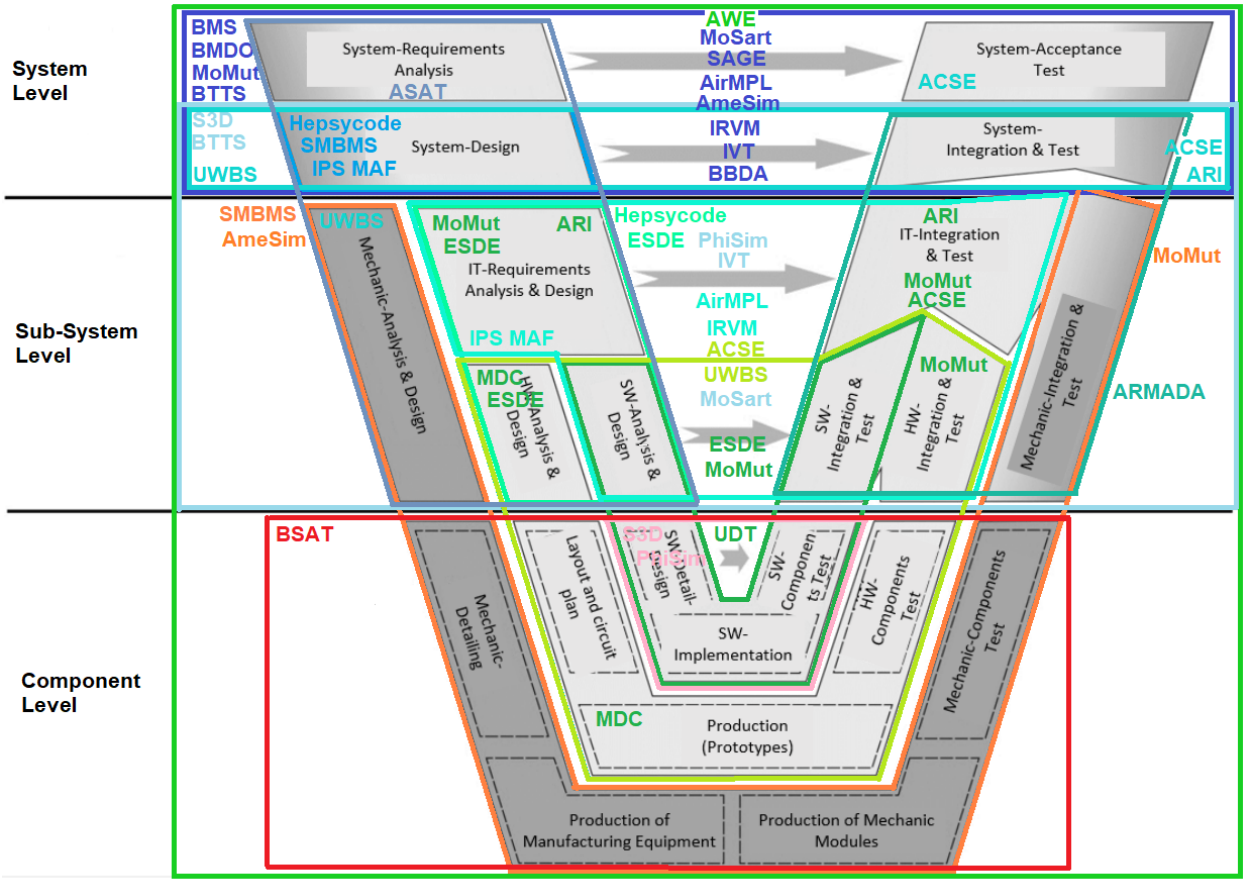


Figure 2 Tools for drone systems.

C4D WP6 will provide improvements in tools covering all the design tasks shown above. The partner, the expected outcome, the final TRL for the technology, the UC in which the technology will be assessed and the design tasks it covers are described in ANNEX I.



BMS: BUT Modeling & Simulation
BMDO: BUT Mission Design & Optimization
UDT: UNIMORE Development Tools
SMBMS: SM Battery Management System
AWE: AIT Workflow Engine
BTTS: BUT Testing Tool Set
UWBS: UWB Batery Simulation
IRVM: IKERLAN Requirement Validation & Monitring
IVT: IKERLAN Validation Toolchain
ASAT: AIT Security Analysis Tool
BSAT: BUT Safety Analysis Tool
BBDA: BUT Big Data Analysis
ARI: AIT ROS Infrastructure
ACSE: ALM Cloud Simulation Environment

Figure 3 C4D coverage of the development V-Cycle.

The different tools provided by the partners are referenced to the V-Cycle of Figure 1so that its potential role along the design and verification cycle and its relation to other C4D tools or external tools from third parties can be better identified. As shown in Figure 3, C4D will deliver a drone system modelling, design and verification framework improving all the steps in the system development V-Cycle. This shows the C4D ambition. The improvements to be achieved will be detailed by the requirements defined by the different Use Cases. These requirements will be described in the next section. In the Figure 3, the list of tools below indicates the acronym used for those tools yet without a name.

It is important to highlight that C4D does not aim to develop a complete drone system development process from scratch. There are many proprietary, commercial and open-source tools already available. The goal of C4D is to take the current state-of-the-art as starting point, take advantage of the background technology provided by the partners and build a comprehensive drone design platform going beyond the state-of-the-art in current practices by improving proprietary tools in conjunction with commercial and open-source tools. Therefore, interoperability is a technical objective to be achieved during tool improvement.

More concretely, the main improvements to be brought to the state-of-the-art in the different stages of the V-Cycle by the C4D WP6 contributors are the following.

1.2.1 System-level design improvements

MoSaRT (ENSMA) will be extended to support drone features that will be proposed in WP3 in order to check the schedulability and timing requirements (end-to-end flows, latencies, etc.) by taking into account the control stability.

HEPSYCODE (UNIVAQ) will be improved to model UAV systems inside the main design flow, while taking into account standard directives and industries' needs.

Papyrus-for-Robotics (CEA) will be extended with support for world models in order to enable autonomous and eventually AI based decisions.

S3D (UNICAN) will be extended to the modeling of Cyber-Physical Systems of Systems (CPSoS) so that it is possible to model, analyse and explore different architectural mapping for complete drone missions.

1.2.2 Sub-system-level design improvements

The battery management system by UWB and SM will provide guidelines for subsystem design for both hardware and software. The components from SM will define needed space and weight of the battery payload and SW interface from UWB will place requirements for the communication interface.

Paparazzi (ENAC) will improve its design (documentation, API, procedures) for easier integration of the elements at the component level.

Simcenter Amesim (SIEMENS) will improve its modeling capabilities concerning the drone flight dynamics performance. This includes new counter-rotating propellers, and improvement on the missions' definition. This will provide more accurate performance results; it will reduce the time needed to create and setup a drone flight dynamics model and it will improve the model reusability.

MDC (UNISS), will be extended to be compliant with the FPGA overlay provided by UNIMORE, to allow for fast prototyping, relieving designers from the burden of specifying the accelerators bitstream and to provide APISto facilitate the adoption/programmability of the overlay architecture.

The ACORDE ESL eSW Design Environment will be improved in its analysis and visualization capabilities, through a more systemization of raw traces obtained from the simulation and the exploitation of existing frameworks, specifically adapted to input, internal and output data related to geo-positioning systems. The work also aims to improve the timing estimation capabilities after a wide consideration tasks of processor and platform estimation frameworks (e.g., Qemu, GEM5 and other proprietary solutions).

The Indoor Positioning System Modelling&Analysis Framework expects to bring an effective tool to design in a holistic approach the anchor deployment, and the anchor and tag firmware for the indoor positioning, specifically suited for the challenges of long indoor infrastructures.

Papyrus-for-Robotics (CEA) will be extended with support for hierarchical components in order to model complex drone architectures.

The battery management system by UWB and SM will provide guidelines for subsystem design for both hardware and software. The components from SM will define needed space and weight of the battery payload and SW interface from UWB will place requirements for the communication interface.

1.2.3 Component-level design improvements

The acceleration infrastructure, resulting of the UNISS-UNIMORE integrated design process, will be coupled to an OS running on GP processors capable of dispatching the tasks and managing housekeeping.

1.2.4 Component-level verification improvements

It will be investigated how dynamic multi-physic models built with Simcenter Amesim (SIEMENS) can support verification and validation of controllers and algorithms for drones navigation, autonomous flight and/or energy management.

AirMPL-Simulator (UNISANNIO) will provide a built-in simulation platform for testing and evaluating path planning algorithms with ground and aerial vehicles in an environment as close as possible to the real scenario (e.g., the demonstrator UC5-DEM10). Compared to the existing solutions, the tool will integrate external software libraries (e.g., OMPL) and simulators (e.g., Gazebo) without requiring any further action, wrapper or glue code.

The integration of simulators such as Gazebo and AIRSIM will become easier in Papyrus for Robotics (CEA).

1.2.5 Sub-system-level verification improvements

Paparazzi (ENAC) will provide a platform to test and validate the components by improving the test procedure against various scenarios reducing by two the time required to prepare a test case simulation for validation.

The AsyncAPI toolchain improves the decreasing detection time of consistency communication problems and facilitate the modeling and documentation tasks.

The battery management system simulator designed by UWB and SM improves verification phase for the usage of battery management system on the real drone. The use simulator will decrease the time needed for battery management system integration to the real device.

1.2.6 System-level verification improvements

Simcenter Amesim (SIEMENS) will develop methodologies to effectively model and simulate drones' performance and to valid. If needed, interfaces with other software tools will be improved with the goal to extend drones' system modeling.

MoSaRT (ENSMA) will help drone architects to find the suitable temporal analysis tests that fit both the hardware and software drone architectures. Transformations to external analysis tools formalisms can be provided.

ReqV and ReqT (UNISS) capabilities will be extended in order to support more expressive logics, such as Signal Temporal Logic. ReqV extension will allow to formulate technical specifications that can be translated in a more expressive logical language, while ReqT will be able to deal with numerical constraints. UNISS will improve performance for MLPs and provide verification algorithms for Neural Networks (NNs). This improvement is necessary for the formal verification of the algorithms developed in the context of what required by UC5-DEM10-FNC-07.

Safety toolchain provides improvements on safety faults detection and reporting time, improving the overall reliability of the system respect to the actual state of the art.

SoSim will enable the simulation and performance analysis of Cyber-Physical Systems of Systems (CPSoS) so that it is possible to simulate, analyse and explore different architectural mappings for complete missions in a short time.

AsyncAPI toolchain allows to define a single point of trust for API specification which fastens the verification tasks related to communication interfaces. It decreases the detection time of consistency problems and will provide a fault injection framework to anticipate and test different situations.

1.3 Design Technology Requirements

As shown in Figure 4, design technologies are traversal to all the drone services value chain. As a consequence, requirements to design and verification tools may come from the component provider, the system integrator and the service provider. All them demand more powerful tools able to shorten de design and verification process ensuring an improvement in design productivity so that both the design time and effort can be minimized even handling more complex components, systems or services. The final goal of the work in WP6 is to satisfy the requirements defined by all the stakeholders in the value chain. Design requirements are identified with the 'DTC' identifier for 'Design Technology'.

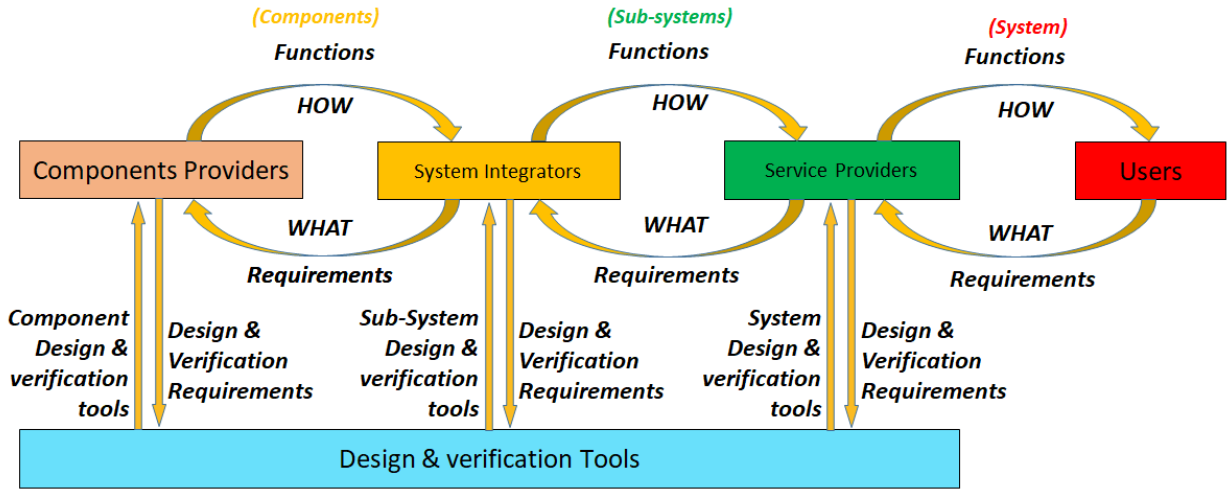


Figure 4 Design Technology requirements along the drone value chain.

The concrete requirements to be satisfied by the tools as a consequence of the WP6 research activity are listed in ANNEX II.

Satisfying all them will ensure to achieve the global project objectives affecting WP6:

Objective O1: Ease the integration and customization of embedded drone systems.

Objective O4: Minimize the design and verification effort for complex drone applications.

2 System Modelling and Code Generation tools

In this section, the tools and methods which have as main objective improving the modeling of the system and the generation of code for different purposes including HW and SW synthesis, are described.

2.1 Modelling & Simulation Tool (BUT)

2.1.1 Tool description

The tool will be used for drone configuration definition and management with special emphasis on high fidelity representation of resulting dynamic models. The designed drone models will then be used in the tool WP6-06 “Mission design and optimization” to enable realistic mission planning capability under operational constrains.

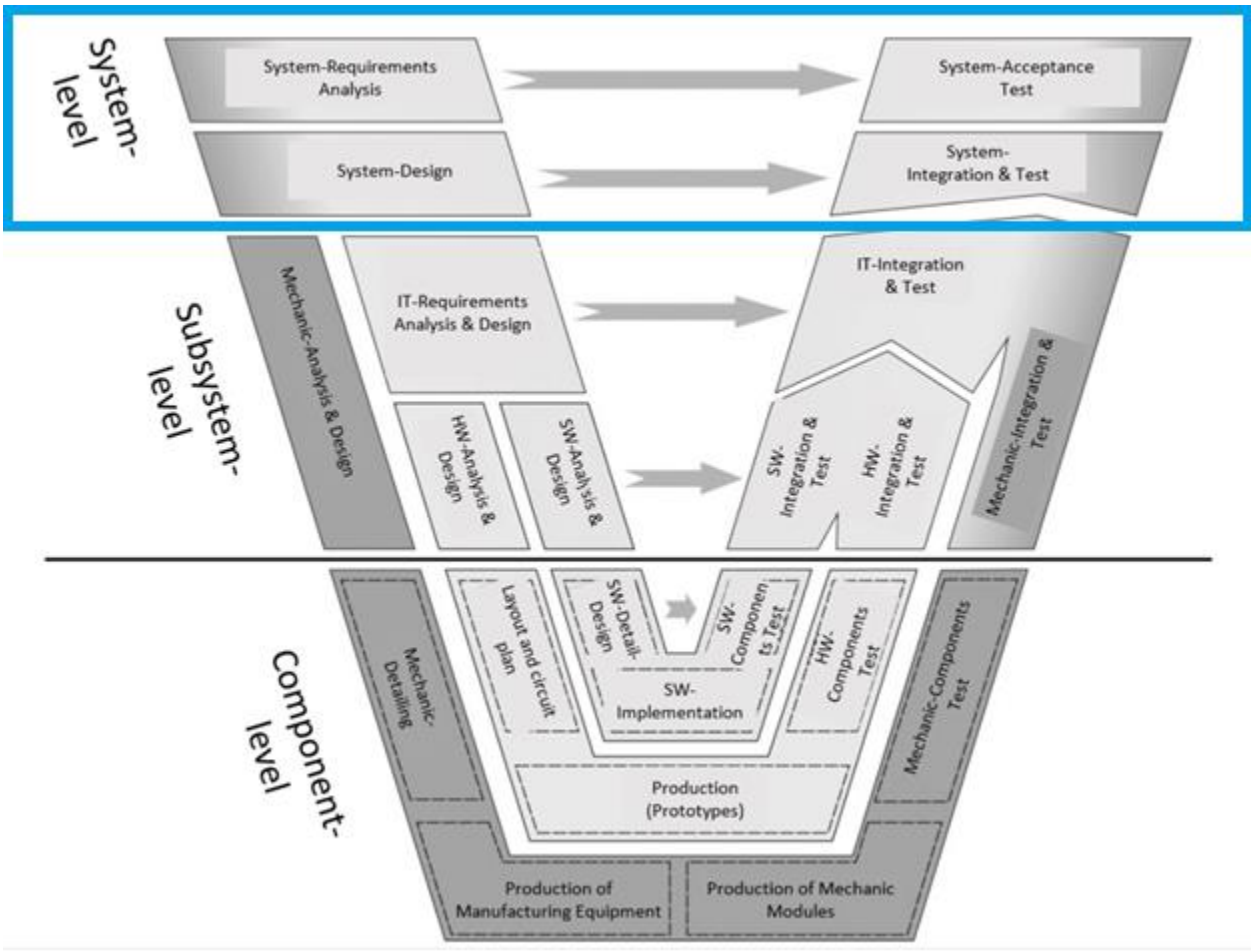


Figure 5 V-Cycle coverage of Modelling and Simulation Tool.

2.1.2 Improvements planned

The tool will be providing high fidelity drone models for a subsequent utilization in the Mission design and optimization tool. The high-fidelity attribute of the tool is an absolute necessity for the computation of viable drone trajectories under real fuel and respective mission constrain scenarios.

2.1.3 Functional and non-functional requirements

UC4-DTC-01	Tool shall provide high fidelity representation of dynamic models

2.2 Mission design and optimization (BUT)

2.2.1 Tool description

A UAV autonomous offshore operation over open seas requires in its mission planning phase consideration of realistic weather data with the additional constraints, like available fuel resources or position of the sun during the target monitoring phase. The toolbox will contain path planning capabilities with a specified target monitoring under imposed constrains.

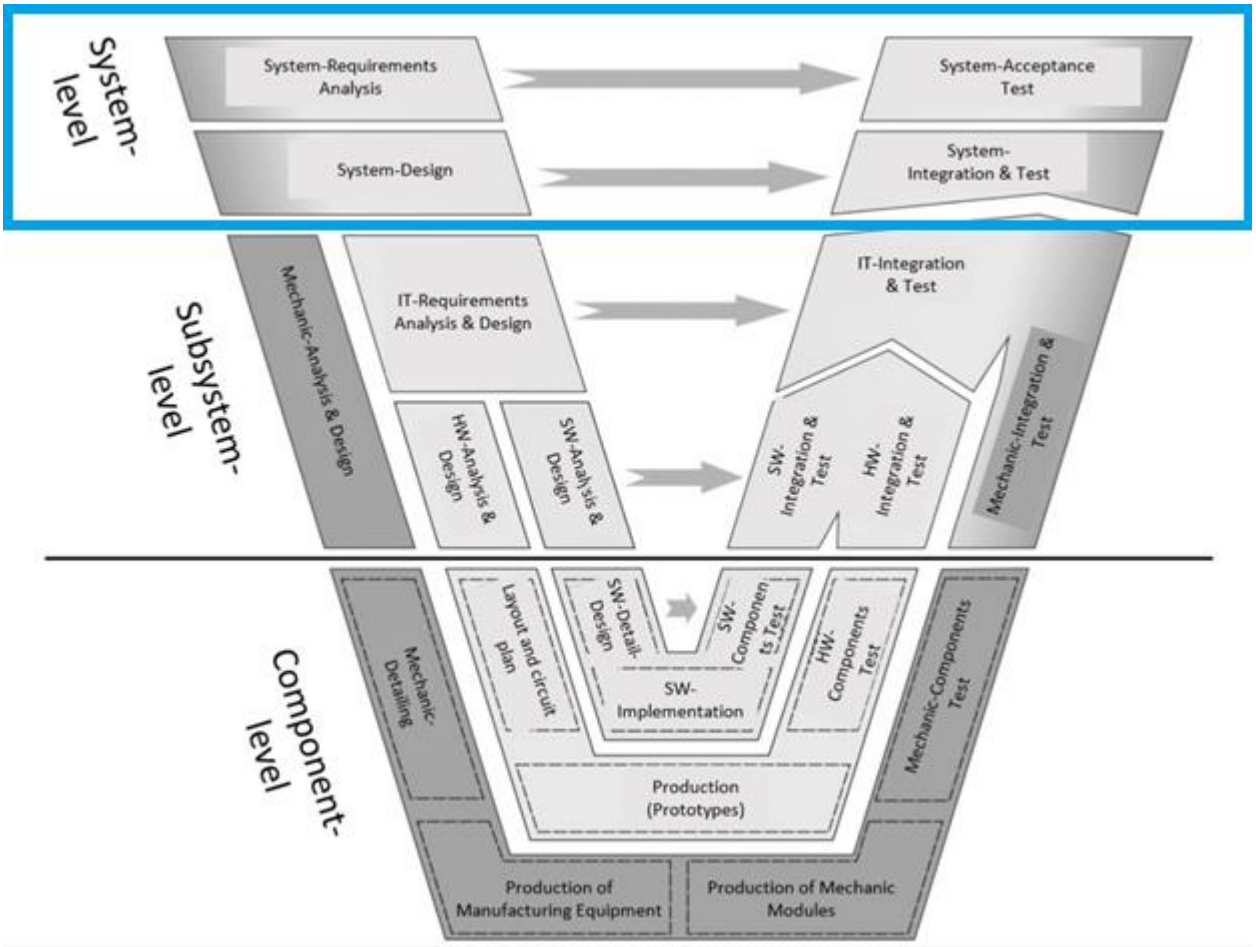


Figure 6 V-Cycle coverage of Mission design and optimization Tool.

2.2.2 Improvements planned

The tool will implement path planning strategies to allow repetitive camera data recordings considering mission specific constrains.

2.2.3 Functional and non-functional requirements

UC4-DTC-02	Tool shall allow mission planning with specified target
UC4-DTC-03	Tool shall consider realistic weather data with the additional constraints, like available fuel resources.

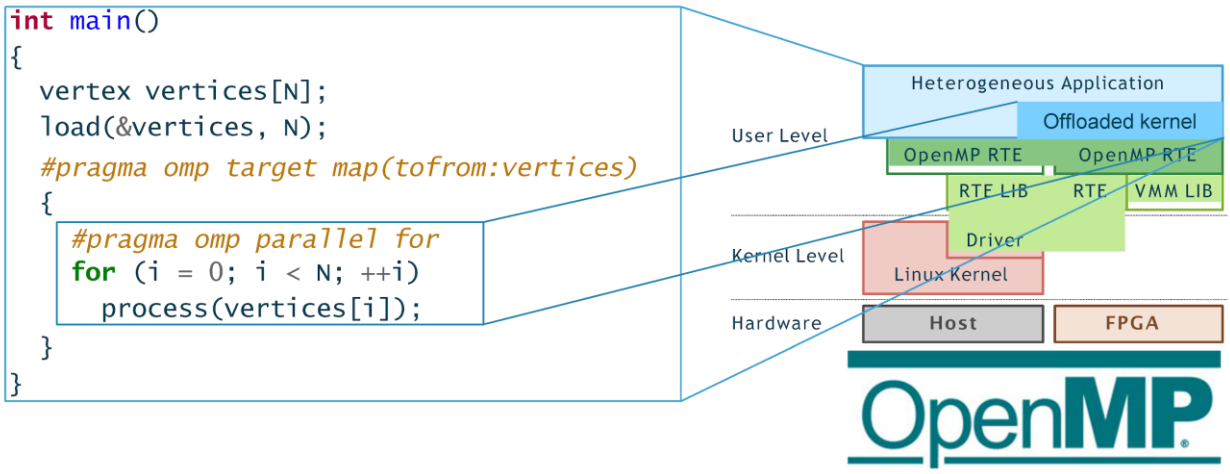
2.3 Application development tools for the heterogeneous on-board computing platform (UNIMORE)

2.3.1 Tool Description

The demand for onboard computational power of modern drones is exponentially growing due to the ever-increasing request for autonomous operation. To satisfy this request, more powerful (in terms of operational throughput) compute platforms must be embedded on the drones, while at the same time maintaining operational constraints related to the power envelope, the form factor, and the interoperability and connectivity with standard drone software stacks.

The main contribution that UNIMORE and UNISS make in WP3 towards this goal consists of a compute platform based on commercial off-the-shelf (COTS), FPGA-based systems-on-chip (SoC). Such platform combines the flexibility of standard CPU processing (the ARM Cortex-A processor integrated in most such SoCs) with a novel open-source FPGA overlay aimed at simplifying the deployment of application-specific accelerators, that we call hardware processing elements (HWPE). This overlay is based on a compute cluster integrating RISC-V cores and low-latency, high-bandwidth shared memory as a standard interface to application-specific HWPEs. More details about this platform can be found in Deliverable 3.1.

However, the more heterogeneous components are integrated into any compute platform, the more complex the integration process becomes in terms of hardware-software interactions, access to shared resources, and diminished regularity of the design. The critical challenges are in the integration with the legacy software components, the programming interfaces, and the management of many heterogeneous components more than in the design of any individual components.

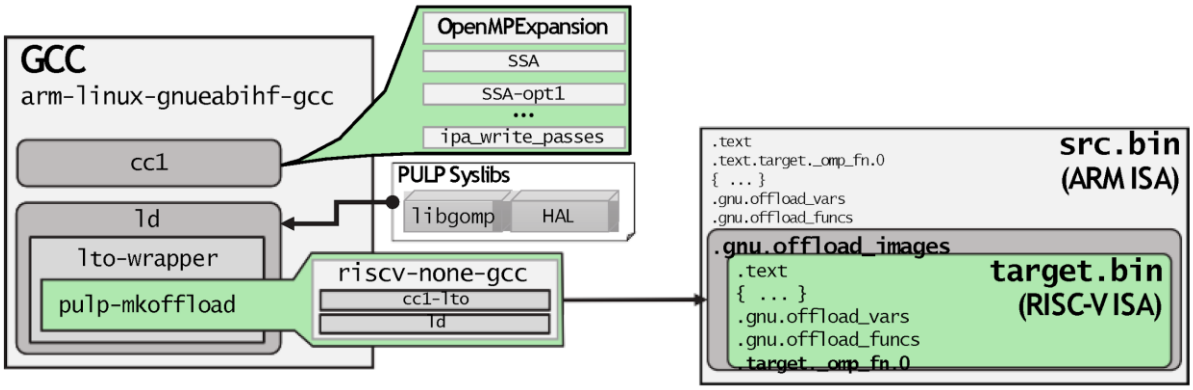


To address this challenge, we propose, as complementary tools to the compute platform: (i) a high-level programming model to offload computation from the ARM CPU to the overlay RISC-V cores, with associated runtime and compiler; (ii) a hardware abstraction layer (HAL) that simplifies the configuration and orchestration of the HWPEs.

The proposed approach is based on a widely adopted shared-memory heterogeneous accelerator model, implemented by the OpenMP v4.x programming interface. The figure below shows how a part of a program running on the main CPU can be simply offloaded to the RISC-V cores deployed as part of the FPGA overlay. To support this model various software libraries and a runtime system will be required.

To pursue the *first* objective - enabling computation offloading from the main CPU to the FPGA overlay - we will start from mature OpenMP implementations and RISC-V compiler backends, like those offered by the GNU GCC toolchain, and develop the following open-source components:

1. GCC extensions to target the proposed FPGA overlay as a compilation backend;
2. OpenMP runtime environment (RTE) extensions for the ARM CPU to communicate with the *compute cluster overlay*;
3. Full-custom OpenMP runtime environment for the RISC-V cores on the FPGA overlay, to enable the use of the OpenMP execution model within the *compute cluster overlay*.



The figure above shows the expected execution flow of the proposed toolchain.

To pursue the *second* objective - simplifying the runtime configuration/reconfiguration of HWPEs and their orchestration - we will develop the following open-source components:

4. An application programming interface (API) to implement a hardware abstraction level (HAL) for simplified HWPE programming (configuration, synchronization, ...) - compared to the low-level register manipulation that is required for their operation.
5. Exploration of OpenMP extensions to further raise the level of abstraction at which HWPEs can be manipulated by the final user.

2.3.2 Improvements Planned

The planned set of components will allow drone application developers to focus on algorithmic aspects of their software, relying on an easy-to-use (directive-based) programming interface for the actual deployment. Stated in other words, from the application developer perspective offloading computations to HWPEs will be no more complex than programming a GPU with OpenCL (or OpenMP itself). The tools will improve the overall accessibility of such special-purpose processing systems using well know programming model interfaces.

The proposed tools will have an impact of the following components of the V-Model:

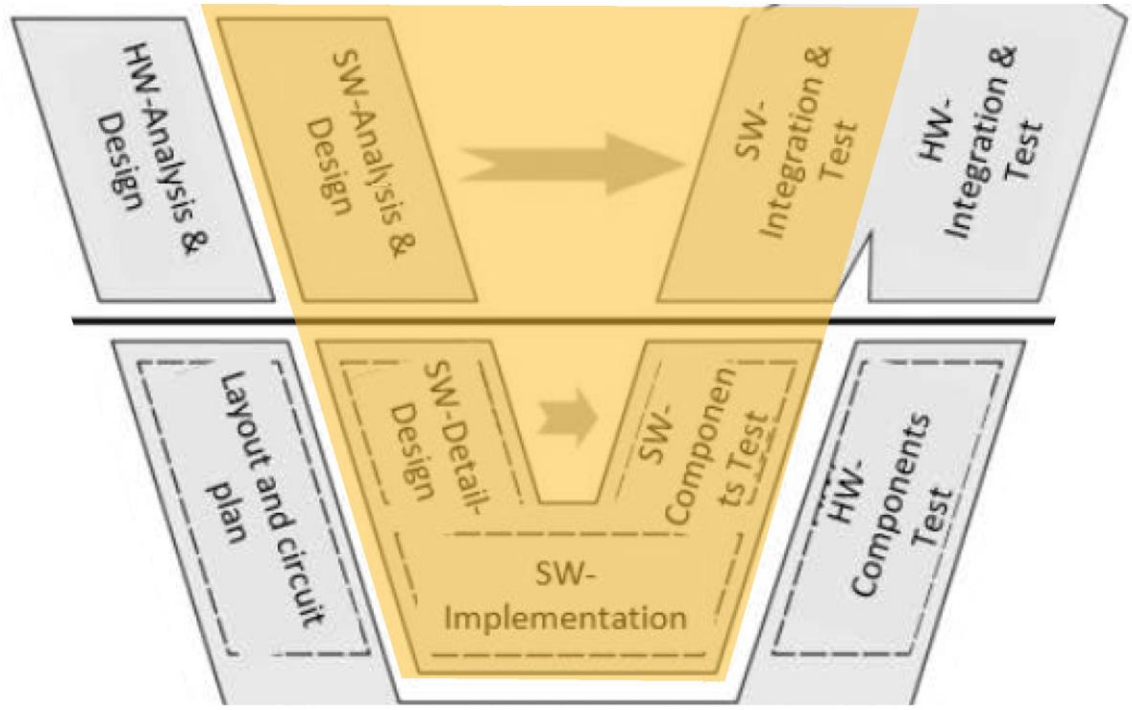


Figure 7 V-Cycle coverage of the proposed programming model.

2.3.3 Functional and non-functional requirements

UC5-DEM10-DTC-04	Accelerators Programming Model SHALL enable offloading capabilities from the host processing system to the HW accelerators mapped on FPGA.
UC5-DEM10-DTC-05	Accelerators Programming Model SHALL enable the configuration of application-specific accelerators mapped on FPGA.

2.3.4 Specific standard and/or regulation requirements

None.

2.3.5 Specific installation and/or functional requirements

It will be required a “standard” GNU/Linux Operating System for the usage of the toolchain and cross-compilation auxiliary tools for the production of the binary and ELF for the compute cluster platform overlay and the Application-Specific Accelerators.

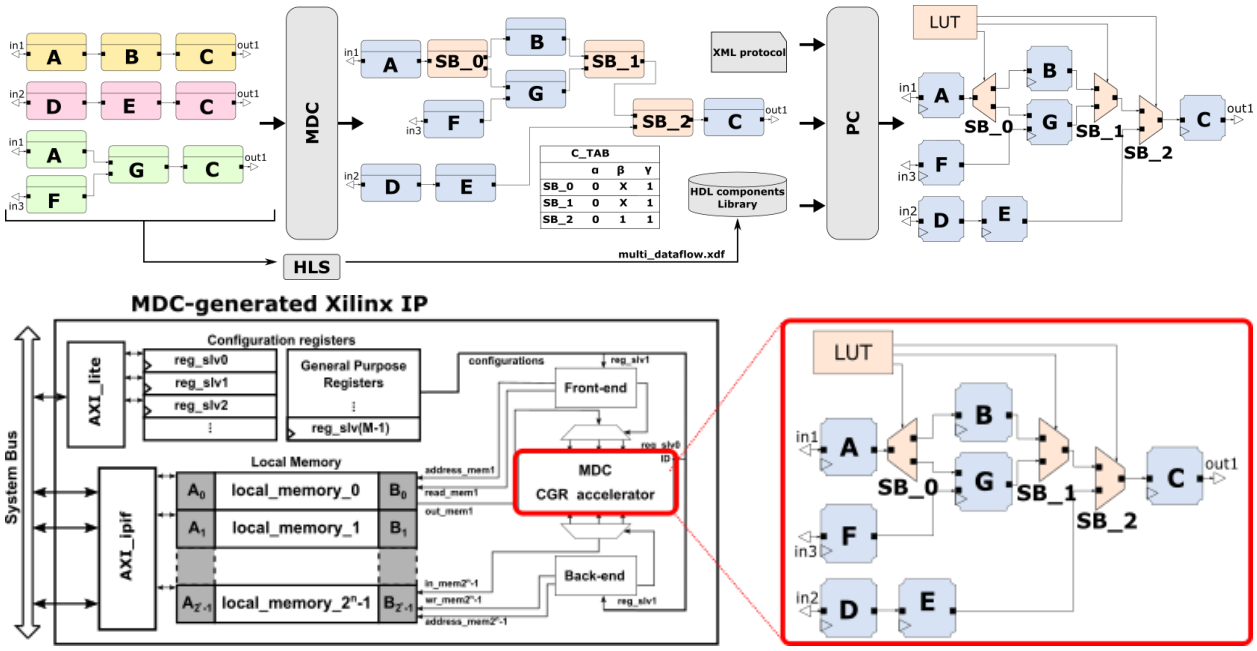
For the deployment, a GNU/Linux Operating System is required as well on the onboard compute companion platform. Such OS will be in charge to execute the host-side OpenMP runtime and communication libraries to the FPGA overlay.

2.4 MDC: Multi-Dataflow Composer (UNISS)

2.4.1 Tool Description

The baseline feature of MDC is the composition of a coarse-grained reconfigurable (CGR) hardware datapath starting from a set of dataflow applications. The baseline feature involves two main components:

- Multi-Dataflow Generator (MDG): it merges together different dataflows into one unique reconfigurable multi-dataflow by the insertion of switching modules named SBoxes. Two different merging algorithms are supported: empiric and Moreano. The former is more suitable for non-recursive dataflows but less optimized than the latter.
- Platform composer (PC): it derives the RTL description of the CGR datapath from the multi-dataflow. It requires the user to define the communication protocol between actors in hardware (XML) and the RTL description of the actors involved in the dataflows (HDL Components Library, HCL).



Among another features MDC provides an automatic coprocessor generation. The Coprocessor Generator automatically embeds the generated CGR datapath into a ready to use Xilinx IP. The user can choose among different options:

- Processor: soft-core (Microblaze) or hardcore (ARM)
- Processor-Coprocessor coupling: Memory-mapped or FIFO-based
- Direct Access Memory Module: enable or not the usage of DMA.

2.4.2 Improvements planned

UNISS will cooperate with UNIMORE to extend an already existing acceleration template. This template is basically an FPGA overlay composed of pre-implemented easily programmable components where application specific computing tasks (i.e. image enhancing, filtering, transformation, cryptography, etc.) can be delegated to custom IPs. The acceleration infrastructure is meant to be coupled to an OS running on GP processors capable of dispatching the tasks and managing housekeeping. Major challenges behind the streamlined adoption of FPGA companion computers in embedded platforms are the following:

- hardware design, customization and integration in the COTS SoC efforts should be minimized.
- from the application developer perspective offloading computations to hardware accelerators should be no more complex than programming a GPU with OpenCL or OpenMP.

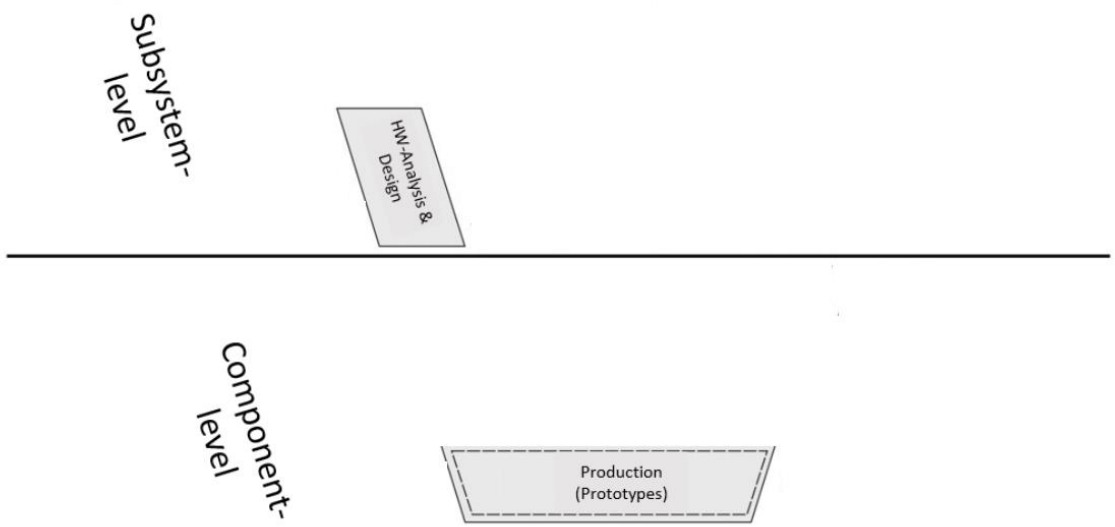


Figure 8 V-Cycle coverage of MDC.

To this end, within **COMP4DRONES**, we intend to adopt and extend the MDC tool, which is meant to be used to create the custom IP within the accelerator to relief designers from the burden of specifying the accelerators bitstream. The MDC Coprocessor Generator will be extended to guarantee the possibility of packing suitable IPs for the given overlay template and to provide proper APIs to facilitate the adoption/programmability of the overlay architecture.

MDC covers the following steps of the V-Cycle of Figure 1.Figure 8

2.4.3 Functional and non-functional requirements

UC5-DEM10-DTC-06	MDC tool will enable the definition of the on board accelerators

2.4.4 Specific standard and/or regulation requirements

N/A

2.4.5 Specific installation and/or functional requirements

https://mdc-suite.github.io/tools/mdc/tutorials/setup

Developers and Executable versions

1) Java JRE 8

Developer version only:

1) Eclipse IDE for Java and DSL Developers
2) Eclipse installation software: Graphiti
3) Apache maven (3.5.0)
4) ORCC - Open RVC-CAL Compiler

2.5 HEPSYCODE: HW/SW CO-DEsign of HEterogeneous Parallel dedicated Systems (UNIVAQ)

2.5.1 Tool Description

Hepsycode is a prototypal tool to improve the design time of embedded applications. It is based on a System-Level methodology for HW/SW Co-Design of Heterogeneous Parallel Dedicated Systems. The whole framework drives the designer from an Electronic System-Level (ESL) behavioural model, with

related NF requirements, including real-time and mixed-criticality ones, to the final HW/SW implementation, considering specific HW technologies, scheduling policies and Inter-Process Communication (IPC) mechanisms. Through the execution of different steps, including a system-level Design Space Exploration (DSE) approach that allows the related co-design methodology to suggest an HW/SW partitioning of the application specification and a mapping of the partitioned entities onto an automatically defined heterogeneous multi-processor architecture, it is possible to proceed with system implementation. The coverage of HEPSYCODE into the **COMP4DRONES** V-Model is shown in Figure 9.

The first step of the HEPSYCODE co-design flow is the Functional Simulation where the system behavioural model is simulated to check its correctness with respect to some testbenches. Testbenches are of critical importance since they have to be as much as possible representative of the possible operating conditions of the system. Such kind of simulation allows to take into account timed inputs (i.e., there is a concept of simulated time), but it doesn't consider the time needed to execute the statements composing the processes, in other words statements (both computations and communications ones) are executed in 0 simulated time. The functional simulation is used to check the correct computation and designers can refine the code at system-level, while focus on other possible metrics that can drive better the designer choice during the whole Product Life-Cycle development flow.

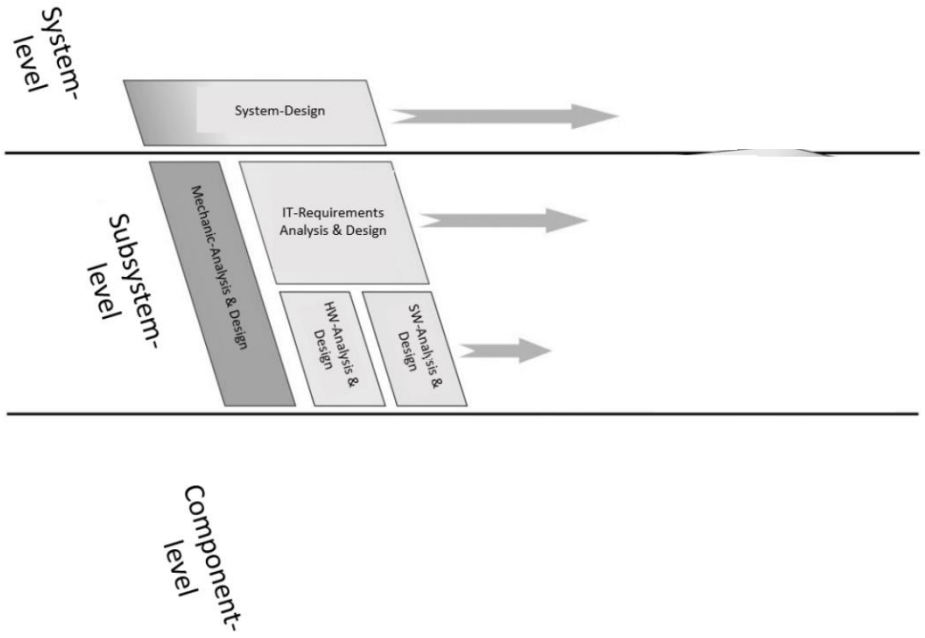


Figure 9 V-Cycle coverage of HEPSYCODE

HEPSYCODE offers an environment and a set of tools that are able to extract also information about system behaviours at different abstraction level. This step aims at extracting as much information as possible about the system by analysing the SBM (Application Model), while considering the available HW components (Platform Model) and the possible use of HPV technologies (Partition Model).

After these metrics' extraction steps, the co-design flow reaches the Design Space exploration step. Starting mainly from Application Model, Partition Model and Platform Model, it includes two iterative activities:

1. *"Search Methods"*, that consider HW/SW partitioning, architecture definition and mapping using an evolutionary algorithm that allows to explore the design space looking for feasible architecture/mapping items suitable to satisfy imposed constraints;

2. *"Timing Co-Simulation"*, that considers suggested mapping/architecture items to actually check for timing constraints satisfaction. This methodology steps will be mapped on the different activities of the HW/SW Co-Design flow.

The *"Search Methods"* is split into two main phases: *Partitioning, Architecture Definition and Mapping Phase 1* and *2* (PAM1 and PAM2). PAM1 provides the partial HW/SW architecture (with the number and type of needed processors), the partitioning between HW and SW components and the mapping between processes and BBs. PAM2 provides the final HW/SW architecture ready to be implemented. PAM2 also finds the number of needed interconnection links (physical links) with a specific topology graph.

Finally, after the PAM activities, the timing simulation helps to check input constraints and to evaluates and compare different solutions applying pareto optimal analysis, using the HEPSIM (HEPSYCODE SIMulator). The whole HEPSYCODE framework and approach is shown in Figure 10.

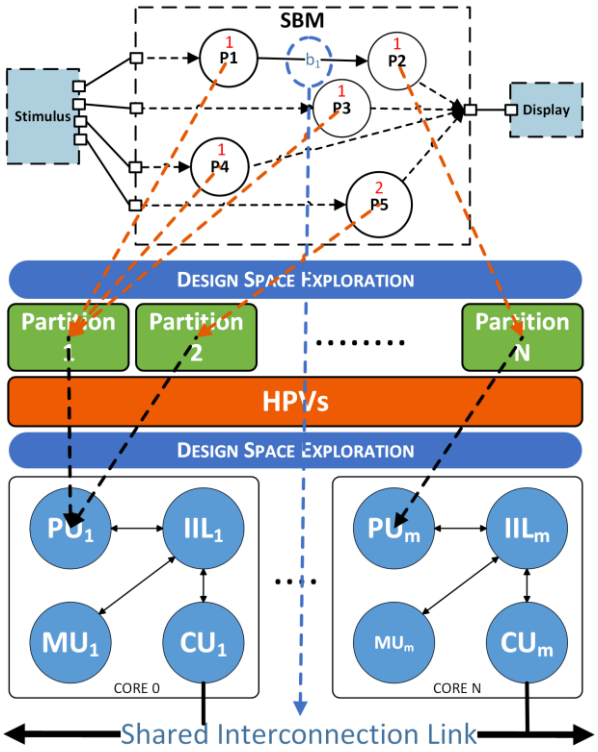


Figure 10 HEPSYCODE Approach

2.5.2 Improvements planned

During the **COMP4DRONES** project, HEPSYCODE tool and methodology will be improved to guarantee:

1. mixed-criticality requirements fulfilment, with particular emphasis on system partitioning, failure/error isolation, timing scheduling, and inter- and intra-task interferences management, improving the Design Space exploration activity during the whole project
2. reduced simulation time and simulator overheads (at least 15% on average simulation time), considering hierarchical scheduling policies
3. a correct characterization of Hypervisors through benchmarking activities

2.5.3 Functional and non-functional requirements

UC5-DTC-07	Design Space Exploration should consider mixed-criticality requirements

UC5-DTC-08	System Simulator should integrate SystemC models
UC5-DTC-09	System Simulator should emulate hierarchical (hypervisor-based) scheduling
UC5-DTC-10	Timing Simulator should consider Hypervisors characterization correctly
UC5-DTC-11	Simulation time should not be dependent on the time granularity related to the events generated by the environment (i.e., the test bench)
UC5-DTC-12	Simulation time should be reduced by at least 15% on average case

2.5.4 Specific standard and/or regulation requirements

The tool uses Eclipse Modelling Framework technologies, and it will be compliant with classical UML/MARTE specification standards as a transformation pattern among several tools.

2.5.5 Specific installation and/or functional requirements

HEPSYCODE uses Eclipse MDE technologies, SystemC custom simulator implementation and an evolutionary genetic algorithm for partitioning activities, all integrated into an automatic framework that drive the designer from first input to final solution. The HEPSYCODE framework consists of a set of reference libraries, scripts and makefiles, eclipse plugins, XML data exchange files, SystemC files, and HW/SW Partitioning And Mapping (PAM) tool.

2.6 ESDE: ESL eSW Environment (ACORDE)

2.6.1 Tool Description

The ACORDE ESL embedded SW Design Environment (ESDE in short), enables a component-based design and validation of algorithms, and the automation of their implementation as firmware to be run on a micro-controller based platform. The framework development was initially started for its application on the design and implementation of firmware for ACORDE GNSS/INS products and developments. The main goals pursued with the tool were:

- The fast development and simulation of embedded SW, as reusable and re-targetable functional components.
- The automation and semi-automation of firmware generation, to avoid the slow and error prone manual translation from algorithm models to the embedded software implementation.
- The speed up of firmware testing, by relying on virtual platform models.

This explains the coverage of the tool on the V-cycle model shown in Figure 11. In its current stage, the modelling, eSW generation and validation tasks are integrated under the same umbrella, an Eclipse-based design environment (as shown in Figure 12).

In its current state, the environment allows the description of the software as a set of communicating SW components (it relies on SystemC language for that). The specification of a GNSS system, integrating specific libraries and exporting output on formats relevant for the analysis in the field (e.g. KML files) is supported. The framework integrates also an eSW generation library which enables, after the development of specific driver dependent channels, a single step, automated embedded SW generation. The executable generated (e.g. an elf file) can be then launched on any of the two versions of the virtual platform models that have been generated so for the aforementioned GNSS-based system.

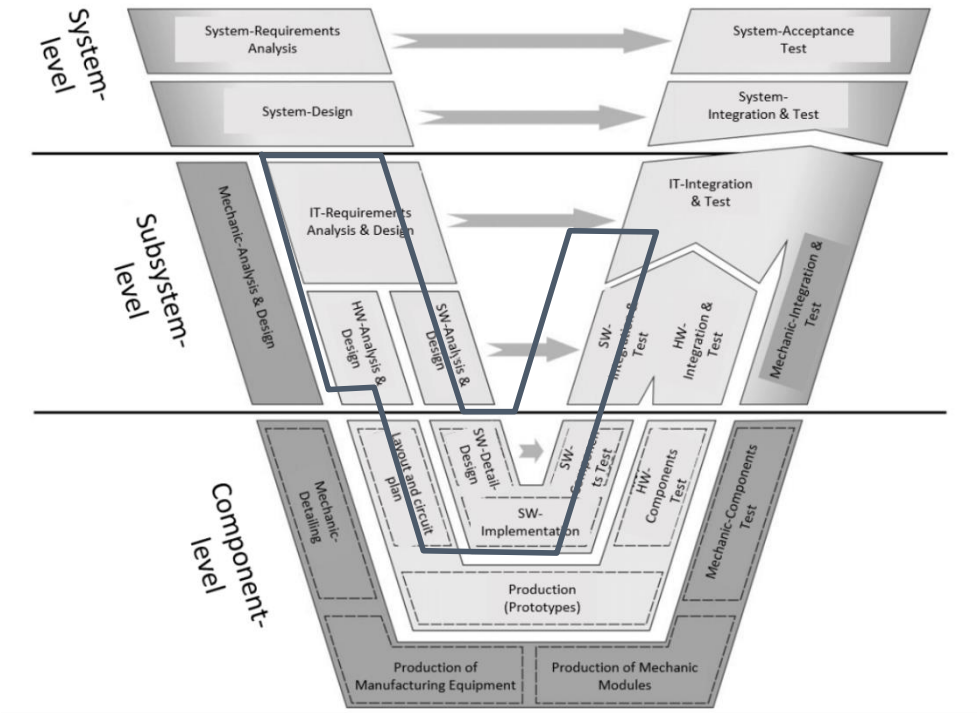


Figure 11 -Cycle coverage of ESDE.

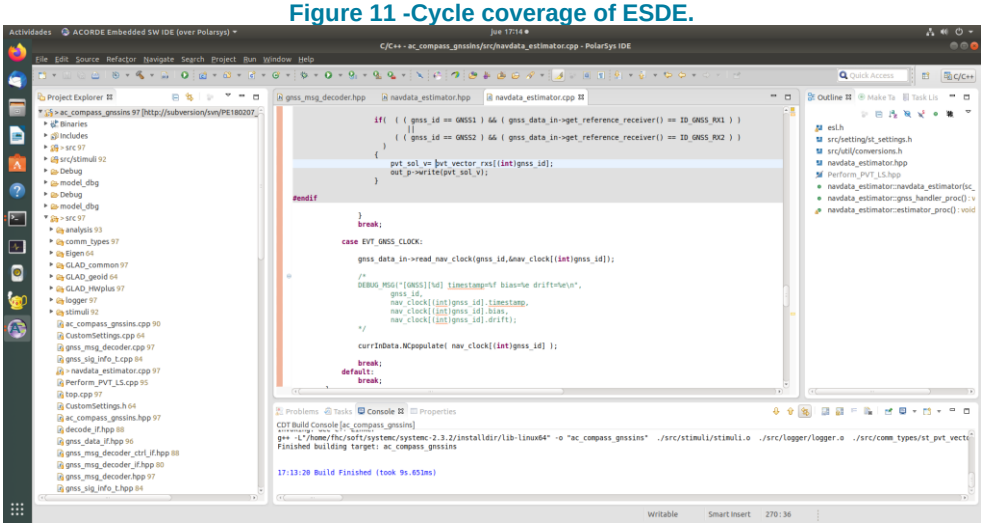


Figure 12 A snapshot of the ACORDE ESL eSW Design Environment

2.6.2 Improvements planned

In **COMP4DRONES** the ACORDE ESDE will be used for further analysis and improvements of the multi-GNSS based attitude estimation algorithms. This includes, the assessment on the impact of using machine learning parts on the algorithm, where so far, a more heuristic approach has been used.

2.6.2.1 Analysis features improvement

This development is expected to serve to refine and improve Octave (Matlab compatible) scripts already used for analysing the outputs. Moreover, there is a specific interest in improving the analysis of raw traces of both inputs and outputs, most of them dumped as text files in custom formats. An initial evaluation of existing Eclipse-based plugins for this is of interest, (e.g., Trace Compass). The goal is to get more user-friendly, graphical, and useful visualization, optionally updated at simulation time. There

are current trends to exploit AI for traces analysis, so the possible application of partner or third-party existing developments will be considered.

Another very specific aspect that it is also of interest is in the improvement of the visualization of data structures in the debugging view. For instance, a template-based like Eigen is of high interest for GNSS system-development, but debugging dynamic data structures under their types is not easy, as the visualization of the actual data requires the navigation of several levels down from the user-level type.

2.6.2.2 Estimation capabilities

Improving the estimation capabilities (e.g. accuracy on time estimates) and simulation speed is on target. In its current stage, the tool potential has been tested against two virtual platform (VP) environments, a commercial, fee-based one and an open one relying on QEMU. However, there is not a dominant optimal solution. An open, free solution with optimal accuracy and control on the processor estimation is desirable. New releases of Qemu report support of processors of interest for ACORDE, not previously supported, but required integration. Moreover, ACORDE will consider a wider emulation&estimation technology scope by considering the possibility to rely on other technologies (e.g., GEM5).

Finally, features with potential exploitation of the framework on other areas of the company beyond GNSS/INS systems, (e.g., control software) might also be considered.

2.6.3 Functional and non-functional requirements

UC2-DEM1-DTC-13	Basic action, i.e. edition, building, and simulation of models; assisted generation of the implementation software (firmware); and firmware verification integrated in a single graphical environment
UC2-DEM1-DTC-14	Integration of version control system
UC2-DEM1-DTC-15	Models based on standard language/procedures
UC2-DEM1-DTC-16	Support of simulatable models, with integrated model of time
UC2-DEM1-DTC-17	Significant improvement of model simulation speed vs Matlab or Octave models (target speed-up=10)
UC2-DEM1-DTC-18	Support of specific libraries for GNSS position&attiude estimation
UC2-DEM1-DTC-19	Support of mechanisms for automating or assisting firmware generation, such the process requires seconds/minutes instead of days/months as it happens on the current manually based generation process
UC2-DEM1-DTC-20	Support for simulation-based verification of the automatically generated firmware, supporting test parallelization.
UC2-DEM1-DTC-21	Simulation based verification faster that running firmware tests on a physical prototype. The objective is speed-up >1 without test parallelization, and scalable verification speed-up with the parallelization of independent tests.
UC2-DEM1-DTC-22	Support of graphical analysis of outputs, e.g. Matlab, Octave, Gtkwave.
UC2-DEM1-DTC-23	Affordable cost of the design environment (<2K€/year)

2.6.4 Specific standard and/or regulation requirements

The following standards are of specific interest for the environment:

- IEEE Std. 1666-2011 (SystemC Language)
- C++ coding standards
- ARM's Cortex Microcontroller Software Interface Standard (CMSIS)

2.6.5 Specific installation and/or functional requirements

So far, the ACORDE ESDE has been tested on a Linux-based (Ubuntu 18.04LTS) OS, as a Virtual machine guest running on Windows 10 host. That Virtual machine requires so far 50GB. Targeting Imperas-based platform models require a third-party tool (Imperas).

2.7 Papyrus for Robotics (CEA)

2.7.1 Tool Description

Papyrus for Robotics is an open-source, Eclipse-based modelling tool for robotic system. It is a customization of the Papyrus UML modelling tool. Papyrus for Robotics conforms to the RobMoSys modelling approach. This implies that the tool supports different views for different roles/stakeholders, for instance a service designer, a component developer and a safety as well as system architect. The different roles are grouped in different *tiers* as shown in Figure 13.

Tier 1 defines the composition structures, tier 2 domain models and tier 3 content produced by eco-system users. Translated to Papyrus for Robotics, the composition structures are the UML meta-model in combination with a Robotics UML profile, the domain models include model libraries that have been defined by domain experts and come in form of pre-packaged model libraries. Tier 3 are concrete component definition, system assembly and high-level behaviour models. The difference stakeholders are shown in the following image. Compared to the V-Shape picture, Papyrus4Robotics is covering the software related aspects of system, sub-system, and component-level aspects.

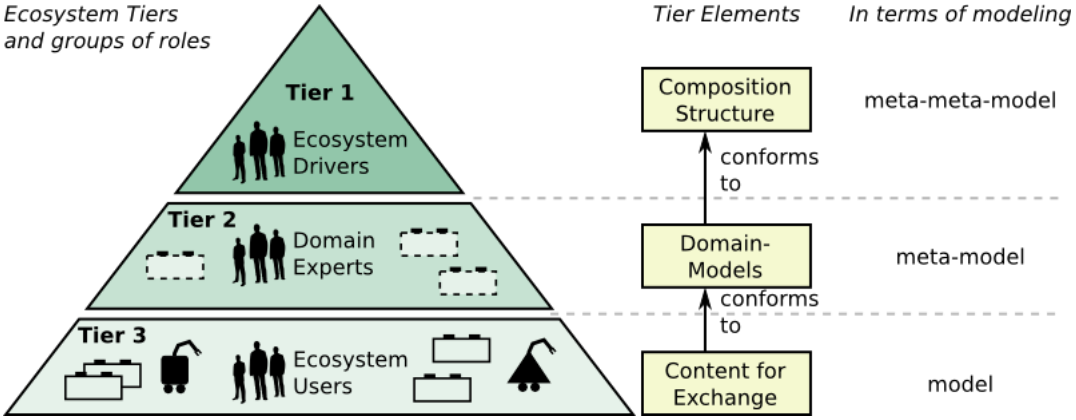


Figure 13 RobMoSys composition structures [RobMoSys wiki].

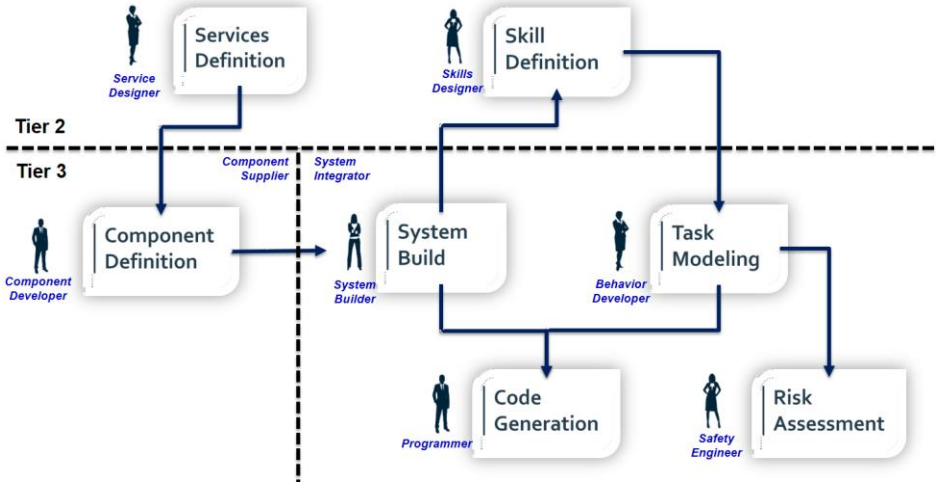


Figure 14 Roles in Papyrus for Robotics.

A user of Papyrus for Robotics can specify a robotic (or drone) system as an assembly of component instances. The components in turn provide or require services via ports. These services are ideally not project specific but standardized by domain experts in order to facilitate the reuse of component definitions. In case of Papyrus for Robotics, the service definitions are part of a set of model libraries that have been obtained from ROS message and service definitions via reverse engineering. This means that a Papyrus for Robotics user has access to a wide range of service definitions that have been established by the community of several years, for instance geometry-related messages or messages needed for navigation. A component instance can be configured, i.e. the configuration parameters of the component definition can be specified for a particular instance. This is shown in Figure 15 – an assembly of component instances, parameters of the PID controller can be configured on an instance level.

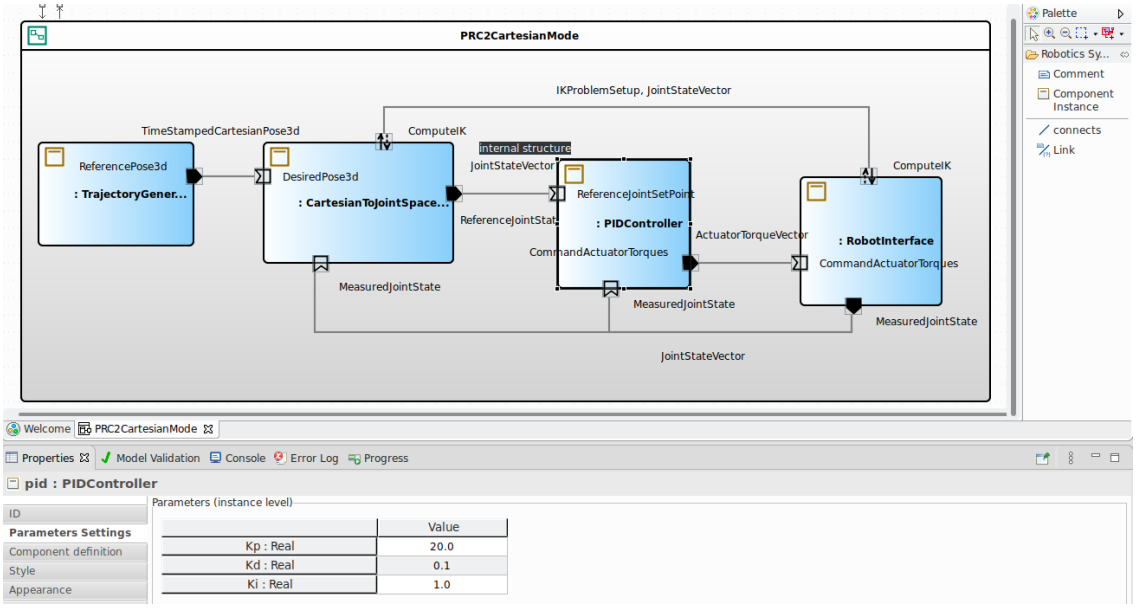


Figure 15 Component assembly and parameter configuration.

2.7.2 Improvements planned

It is planned to have a specific library for the drone's domain, notably service and component definitions that are specific for the drone domain. These can include AI-based components. It is also planned to improve usability of the tool.

2.7.3 Functional and non-functional requirements

Requirements can be specified in form of SysML requirements. These can be linked with other model artefacts to verity that a certain coverage. For instance, a timing requirement can be linked with a service offered by a component.

UC3-DEM02-DTC-24	Reduce the effort required for developing a system component.
UC3-DEM02-DTC-25	Ease the integration and configuration of the system different components.

2.7.4 Specific standard and/or regulation requirements

The tool complies with the RobMoSys modelling approach. With respect to the task-based hazard and risk analysis, compliance with safety norms such as IEC 61508-3 or ISO 13849-1.

2.7.5 Specific installation and/or functional requirements

Papyrus for Robotics can be installed from:

https://www.eclipse.org/papyrus/components/robotics.

It comes as a prepacked Eclipse version (RCP) for the 64bit variants of Windows10, Linux and MacOS. There is also an Eclipse update-site that enables the installation in any (recent) Eclipse installation. It is planned to add the tool to the Eclipse Market place. Besides of the tooling, a ROS2 distribution such as Eloquent or Dashing needs to be installed separately.

2.8 S3D: Single-Source System Design Framework (UNICAN)

2.8.1 Tool Description

Modeling is task consuming and error-prone. S3D ensures minimal modeling effort while achieving:

1. maximal reusability to different projects, thus minimizing the development effort in a product line as well as product up-dating and maintenance,
2. maximal flexibility to optimize the model to any concrete system requirements,
3. maximal adaptability to any industrial design flow by supporting any concrete tool required along the design process.

From the System Model, automatic Model-to-Model (M2M) and Model-to-Code (M2C) generators will extract the information required to support any particular design step. So, among other tools that can be integrated as well, the following are provided:

MAST supports schedulability analysis,

VIPPE supports simulation and platform-aware performance estimation,

eSSYN supports SW synthesis.

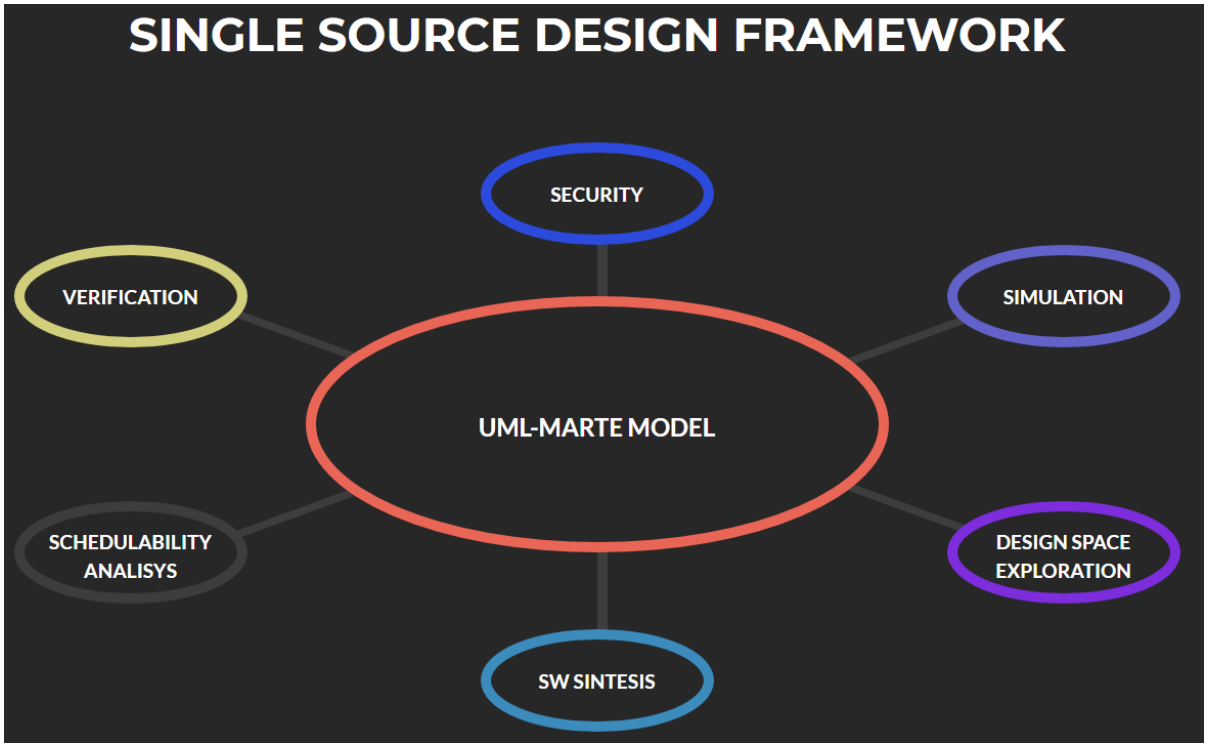


Figure 16 S3D Framework structure.

S3D proposes a component-based approach where components are selected from component's libraries. An example of such a library is shown in Figure 17. Library components can be used as many times as required. In this way, S3D supports reusability. Components are described providing all the relevant information about the component needed by the different tools:

1. Interface data types,

2. Required and provided services,
3. Functional code(s),
4. Verification code.

Components can be integrated to form sub-systems. As the system and sub-system models has the same structure than a component, they can be integrated in a library once finished. In this way, System-of-Systems modeling is supported.

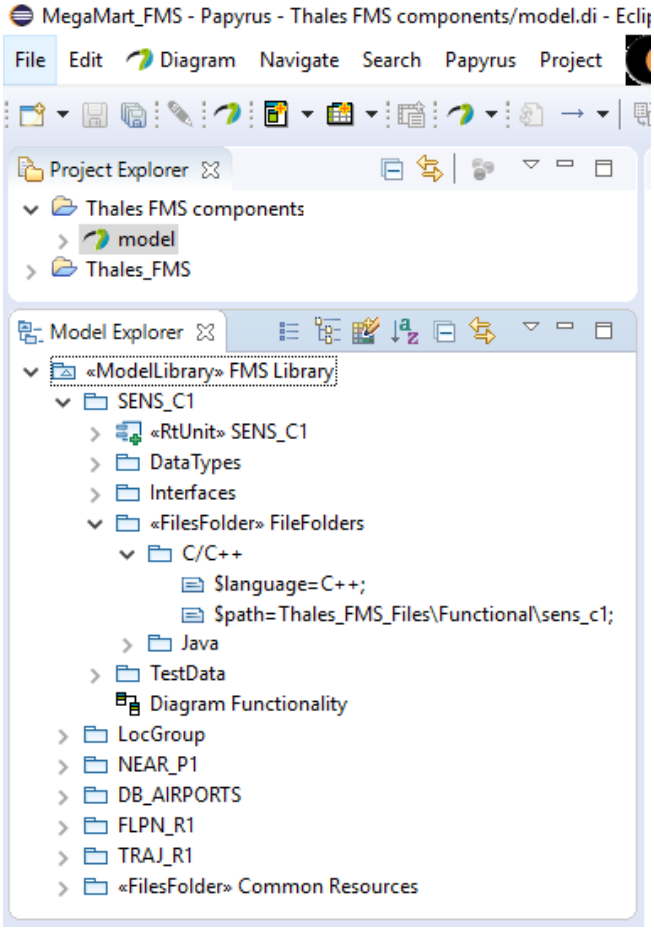


Figure 17 An S3d Library of components.

2.8.2 Improvements planned

S3D will be extended to the modeling of Cyber-Physical Systems of Systems (CPSoS) so that it is possible to model, analyse and explore different architectural mapping for complete missions. From the CPSoS model, SiL, MiL and HiL models will be automatically produced. The final implementation will be also automatically generated using the ESSYN tool.

eSSYn is a software synthesis tool for embedded systems. With eSSYN the embedded system designers can generate directly from the S3D model, a complete set of target binaries for an embedded application in a matter of minutes. The tool significantly shortens development time and allow for future reuse.

In order to use eSSYN, system designers need to provide:

- A software component based model of the application. As part of this model designers need to provide functional code (C/C++) for the components.
- A model for the hardware platform that specifies the available resources (mainly number and type of cores and operating systems).

- Mapping of software components and cores.

At the end of the project, the coverage of the V-Cycle of Figure 1 by the combined use of S3D+eSSYN+SoSim will be that shown in Figure 18.

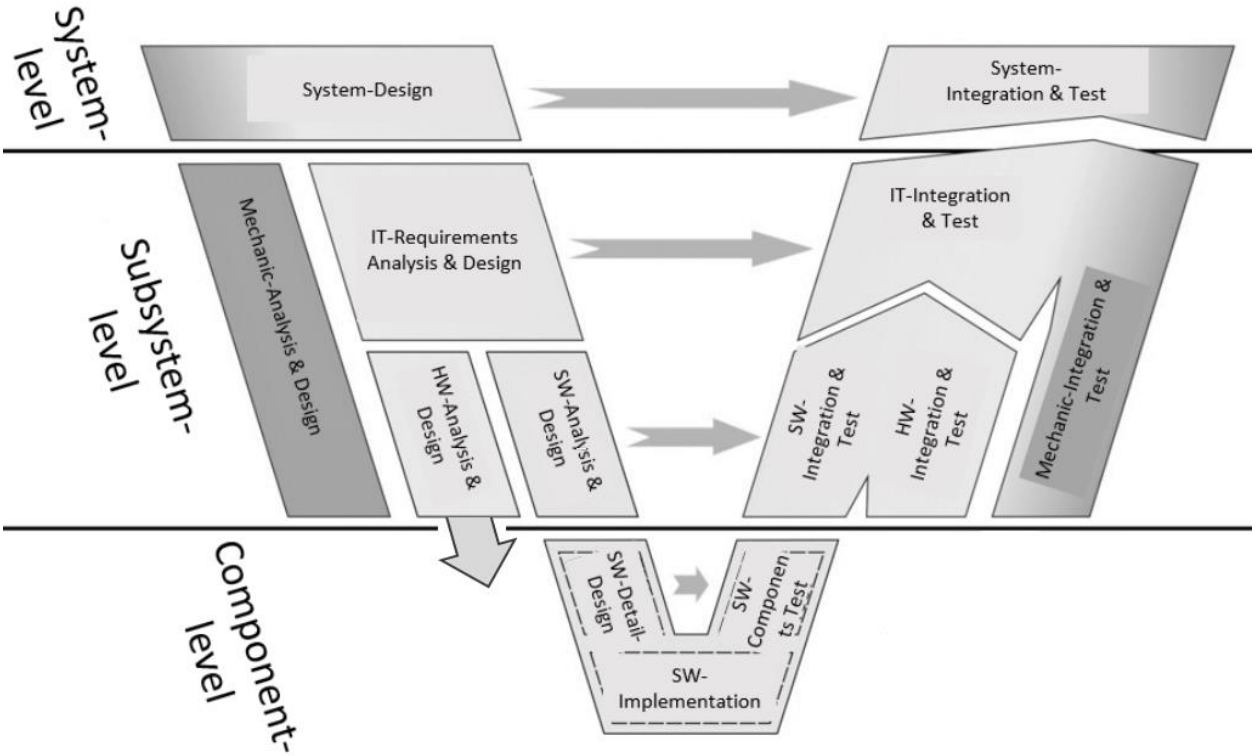


Figure 18 V-Cycle coverage of S3D + eSSYN + SoSim.

2.8.3 Functional and non-functional requirements

UC3-DTC-26	S3D will be able to model a complete drone mission.
UC3-DTC-27	From the S3D model a simulation model for the drone mission will be generated.
UC3-DTC-28	The verification cycle using the S3D Model in the Loop (MiL) technology will reduce the model-simulation cycle in more than 10%.

2.8.4 Specific standard and/or regulation requirements

The framework uses UML/MARTE as background meta-model. Nevertheless, this is made transparent to the user.

2.9 Design tool for drone mechanical parts and subsystems development for autonomous drone battery management system (SM)

2.9.1 Tool Description

Set of tools to improve design of drone components and accessories in environment of CAD and CAE in example SolidWorks system. Tool and component library will allow to adapt a non-specific drone to able to be used with DronePort technology (autonomous battery management). DronePort will consist of two main parts. The first part is related to the drone (battery case, power and data connectors, interface module, etc.), the second part is related to the DronePort charging station itself. The tool will

provide the user to determine components to be used in a specific user case. In example: size on connecters, number of contacts, battery case, interface module, battery gripper, charging station size, number of charging slots, etc. The Library will consist of a CAD and HW electronic components to adapt users drone to be used with DronePort technology.

This tool will be used as a design tool during development in WP3.

The coverage of the simulator into the **COMP4DRONES** V-Model is shown in Figure 19.

2.9.2 Improvements planned

The goal is to enable a user drone integration with DronePort technology. DronePort technology allow a true long-term autonomous drone operations.

2.9.3 Functional and non-functional requirements

UC4-DTC-29	The tool shall lead the developer through the design process of DP integration
UC4-DTC-30	The design library should unify and simplify the design of DP battery system hardware

2.9.4 Specific standard and/or regulation requirements

There are no specific standard and/or regulation requirements.

2.9.5 Specific installation and/or functional requirements

Current version of library support only the SW Solidworks. The support for universal format CAD data is in development.

The library does not have strict requirements for the HW.

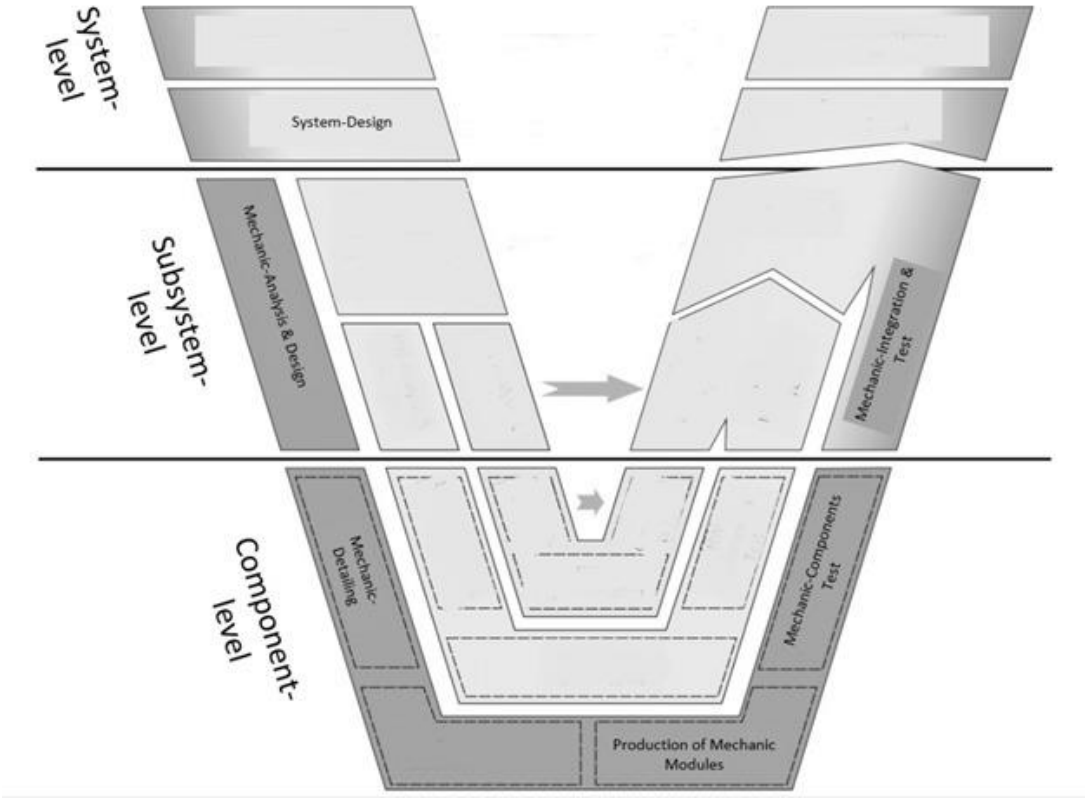


Figure 19 V-Cycle position for the SM drone battery management system.

3 System Validation and Verification tools

In this section, the tools which have as main objective the validation and/or verification of the design process, are described.

3.1 Workflow engine (AIT)

3.1.1 Tool Description

AITs workflow engine has the goal to facilitate validation, verification (V&V) and certification of safety relevant systems in a modular (component based) manner. The safety case is the central output of the workflow engine. It summarizes the information of the V&V process and provides a basis for the certification of the artefact under test (AUT). The validation plan (v-plan) consists of the requirements for the AUT as well as the V&V activities which are necessary in order to satisfy those requirements. A V&V activity is the application of a V&V method by means of an appropriate V&V tool. Positive results of the V&V activity are used to establish evidence for the requirements, while negative results are fed back to the developer team.

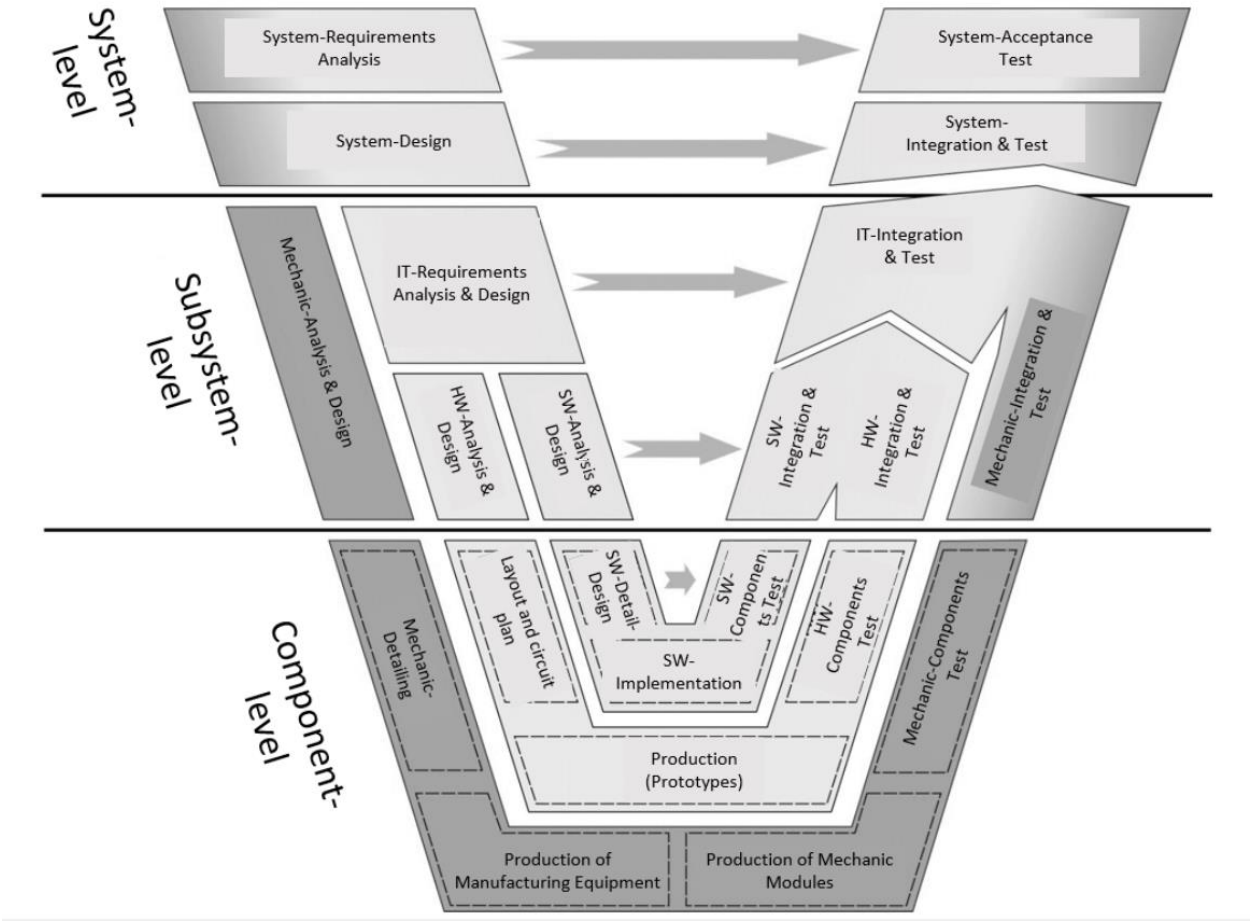


Figure 20 Intended coverage of the AIT workflow engine.

3.1.2 Improvements planned

We plan to extend the workflow engine towards security and enable a combined safety and security process. Similar we also plan to identify and develop a sort of “security” case which supports a modular security argumentation.

3.1.3 Functional and non-functional requirements

DEM9-DTC-31	Development workflow

3.1.4 Specific standard and/or regulation requirements

While there are no specific standards or regulations for the tool itself (not in the Proof of concept variant) the developed workflows will be based on safety and or security standards.

3.1.5 Specific installation and/or functional requirements

This is still under definition.

3.2 MoMuT protocol testing (AIT)

3.2.1 Tool Description

MoMuT is a model-based testing tool addressing functional and non-functional test-case generation from behaviour models. It can work from several different input modelling languages (e.g. UML and domain specific languages).

Non-functional aspects that can be addressed are:

- safety invariants
- robustness testing
- performance

3.2.2 Improvements planned

Drone communication needs to be secure. At the same time, communication must not consume too much power, affecting the operative time of the drone. These two aspects – security and performance – are often in conflict with each other.

The existing base tool will be used to develop and explore new approaches for performance-sensitive security protocol testing. This allows to verify if the intended trade-offs between security and performance are actually implemented as expected. It will possibly include combining it with (binary) program analysis, providing a framework for assuring specific security aspects for drone communication.

3.2.3 Functional and non-functional requirements

DEM9-DTC-32	Model-based testing for drone communication

3.2.4 Specific standard and/or regulation requirements

Not known.

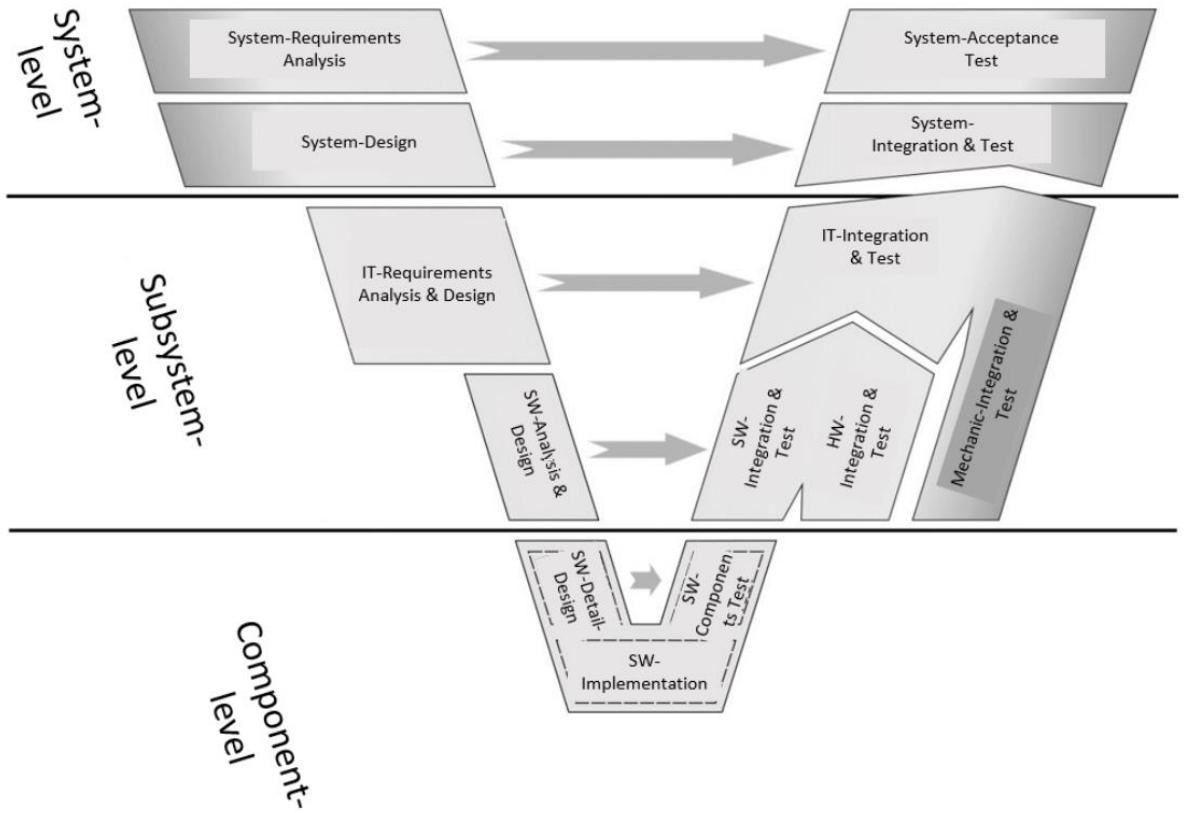


Figure 21 Intended coverage of MoMuT protocol testing

3.2.5 Specific installation and/or functional requirements

Depending on the concrete development, the testing tool might be limited to use on Linux systems. This does not limit the system under test itself.

3.3 Testing Tool Set (BUT)

3.3.1 Tool Description

FIT BUT will provide a tool that based on a known trajectory (locations, speeds, and orientations in time) provides a simulated camera input for further processing by image/video processing software. This tool is intended for simulated HDR multi-exposure assembly, potentially also for multi-RGB with filters) multi-exposure multispectral image assembly, etc. It is assumed that the tool will help building the image/video processing software that will properly register the images and will avoid need for unnecessary extensive image data acquisition e.g. in presence of wind, bad weather, etc.

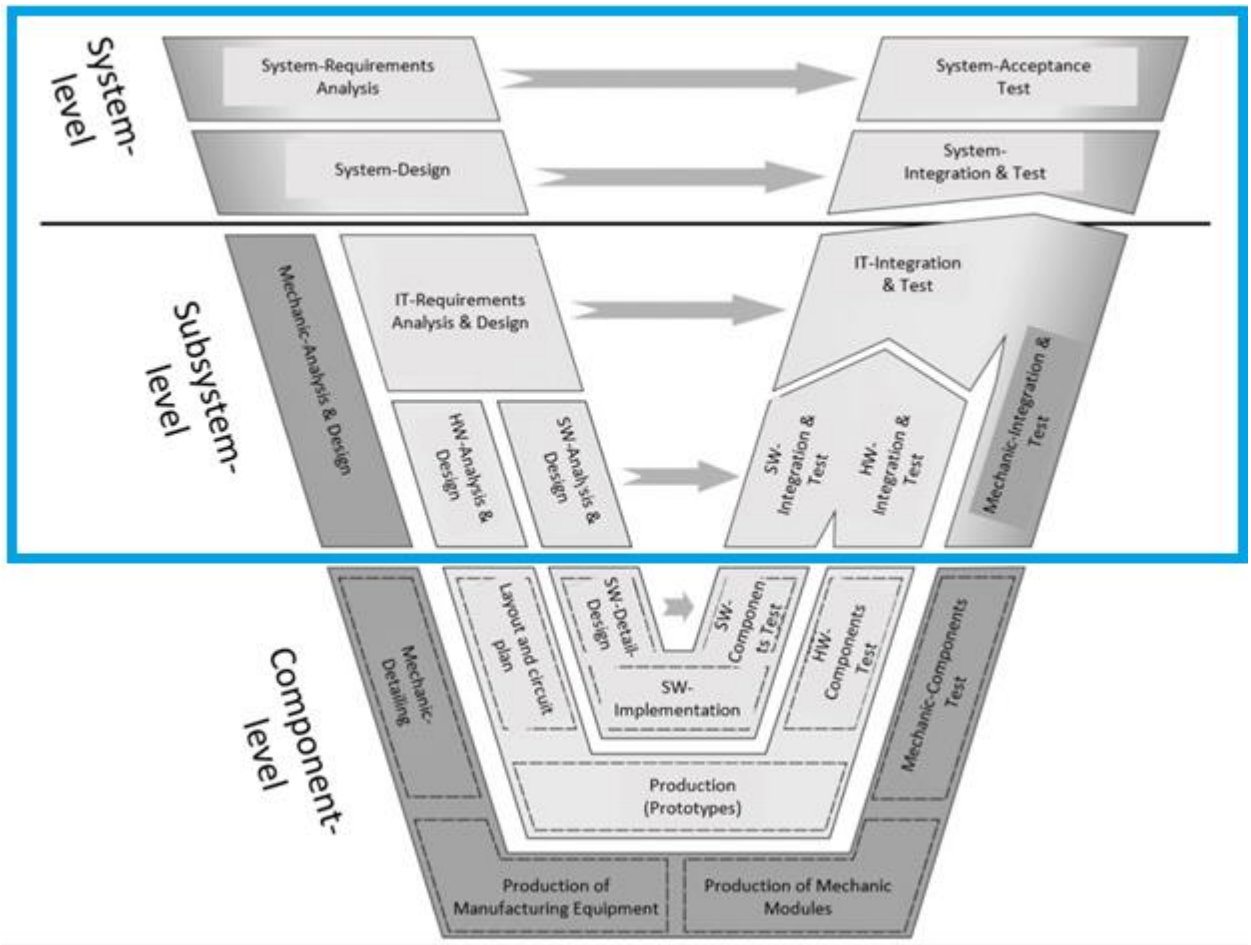


Figure 22 V-Cycle coverage of the Testing Tool Set.

3.3.2 Improvements planned

The tool will be enhanced with the following functionality - weather and wind effects will be added.

3.3.3 Functional and non-functional requirements

UC4-DTC-33	The simulator shall simulate weather and wind effects.
UC4-DTC-34	The simulator shall allow capturing simulated HDR for HDR multi-exposure assembly.

3.4 ROS/Gazebo modules for autonomous drone battery management simulation and validation (UWB)

3.4.1 Tool Description

Drone battery management simulator is a simulation tool based on the Gazebo environment simulator interfaced via ROS. Drones are controlled using PX4 flight controller in simulation mode PX4 SITL.

Gazebo offers the ability to accurately and efficiently simulate populations of drones in complex indoor and outdoor environments. It contains robust physics engine, high-quality graphics, and convenient programmatic and graphical interfaces. Gazebo is open source software with active community (see http://gazebosim.org).

The employed open source flight control software PX4 for drones and other unmanned vehicles can perform various basic flight controls and mission tasks (see https://px4.io).

This simulator will implement virtual device called Droneport (DP). It is autonomous station with defined communication interface via MAVLink messages. The DP is able to guide drone for landing, change / charge its battery and return it back to action. The drone port will be extended with DP traffic control which will autonomously monitor several drones and takes care about the air traffic near the DP.

The simulator will be used as a prototyping and validation tool during development in WP3 and WP4.

The coverage of the simulator into the **COMP4DRONES** V-Model is shown in Figure 23.

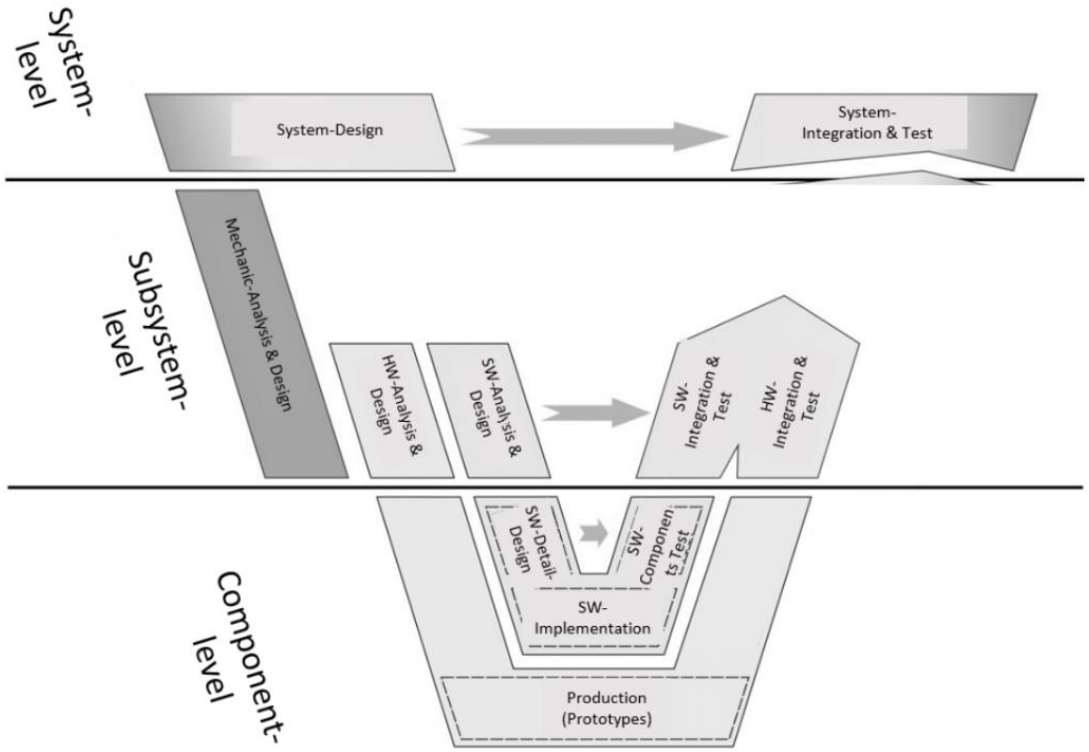


Figure 23 V-Cycle coverage of the autonomous drone battery management simulation and validation.

3.4.2 Improvements planned

The goal is to developed the ROS/Gazebo extensions for simulation of air traffic and mission control in vicinity of Droneport. These extensions cover the using of droneports as a charging station and their traffic management.

3.4.3 Functional and non-functional requirements

UC4-DTC-35	The simulator shall emulate refuelling process
UC4-DTC-36	The simulator shall communicate using MAVLink messaging protocol

3.4.4 Specific standard and/or regulation requirements

The gazebo based droneport simulator will utilise MAVLink messaging protocol so the communication must comply with this message format.

3.4.5 Specific installation and/or functional requirements

Current version of the Gazebo support only the OS Linux. The support for OS Windows is in development. The simulation does not have strict requirements for the HW. So standard office computer should be sufficient.

3.5 Paparazzi UAV System (ENAC)

3.5.1 Tool Description

The Paparazzi UAV system is an open-source hardware and software system designed for experimentation with mini and micro air vehicles.

It is supporting fixedwing aircraft, multicopter and recently ground rovers. It also features a unique flight plan system that makes it easy to create very complex fully automated missions.

The system is designed for multi-UAV operation and simulation with a realistic flight dynamic model. Several autopilot boards are supported, including some third-party solutions like pixhawk boards or Parrot drones.

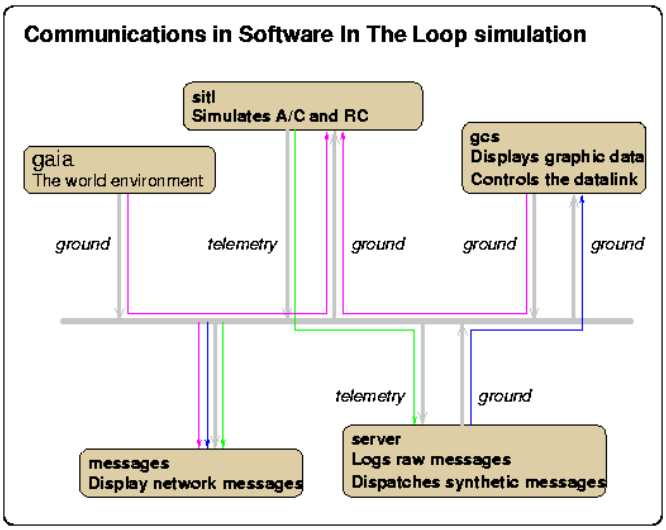
A flexible middleware allows to easily connect external tools and interact with the system for both real and simulated flights as show on the figure below.

The coverage of this tool regarding the V-Model is focused on the subsystem level.

3.5.2 Improvements planned

The objective for this tool is to improve the validation of code modifications with an automated testing procedure including specific compilation, running various static analyses tools and automated simulation.

The internal validation will focus on improving the code coverage with automated compilation and execution of static analyses. The current coverage, is about 20 to 30 % (specific compilation) and we expect to reach at least 60 to 70 % of coverage to match expectation of SC4.1.



A specific effort will be made to deploy a simulation framework that can be used from other partners to simulate multi-UAV operations with a realistic flight model in order to validate external packages. Eventually hardware in the loop simulations are also possible. The objective is to reduce by two the time required to prepare a test case simulation for validation to match expectation of SC1.1.

3.5.3 Functional and non-functional requirements

UC3-DTC-37	Reduce the effort required for integrating a component to the system, either tightly or loosely coupled
UC3-DTC-38	Enable the simulation of several heterogeneous drones with realistic flight models and wind conditions
UC3-DTC-39	Provide control and navigation algorithms, with complex flight plan routines if necessary
UC3-DTC-40	Perform automatic compilation and verification of the code base to provide a safer tool

3.5.4 Specific standard and/or regulation requirements

Configuration files are mostly using the XML format. The communication library, called PPRZLINK, is specific to the project, but is very similar to MAVLINK v1, for which a very limited support (basic monitoring) is also possible.

3.5.5 Specific installation and/or functional requirements

The installation procedure is described in the Paparazzi wiki, which is the main documentation of the project: http://wiki.paparazziuav.org/wiki/Installation.

The recommended OS is Ubuntu 18.04, however the recent integration of WSL2 in Windows10 makes it possible to perform the Linux installation on a Windows station. In addition, a VirtualBox image will be made available for a rapid installation for testing.

3.6 AirMPL-Simulator (UNISANNIO)

3.6.1 Tool Description

The AirMPL-Simulator (Aircraft Motion Planning Simulator) provides a Software-in-the-loop (SIL) simulation platform (see Figure 24) where path planning algorithms, such as those provided in WP4, can be tested and evaluated.

AirMPL-Simulator allows to simulate the UAVs (Unmanned Aerial Vehicles) and UGVs (Unmanned Ground Vehicles) dynamics in a scenario quite close to the demonstrators (e.g., UC5-DEM10) by using the Gazebo robotic simulator along with the RotorS ROS package. Its modular architecture helps the integration of external software modules (i.e., ROS packages) provided by the consortium or from open-source libraries, such as OMPL (Open Motion Planning Library).

The platform is designed to be used as a test bench before real experiments, allowing to set-up different simulation scenarios where simulated autonomous agents can carry out complex missions which could be time consuming, expensive and risky in the real world.

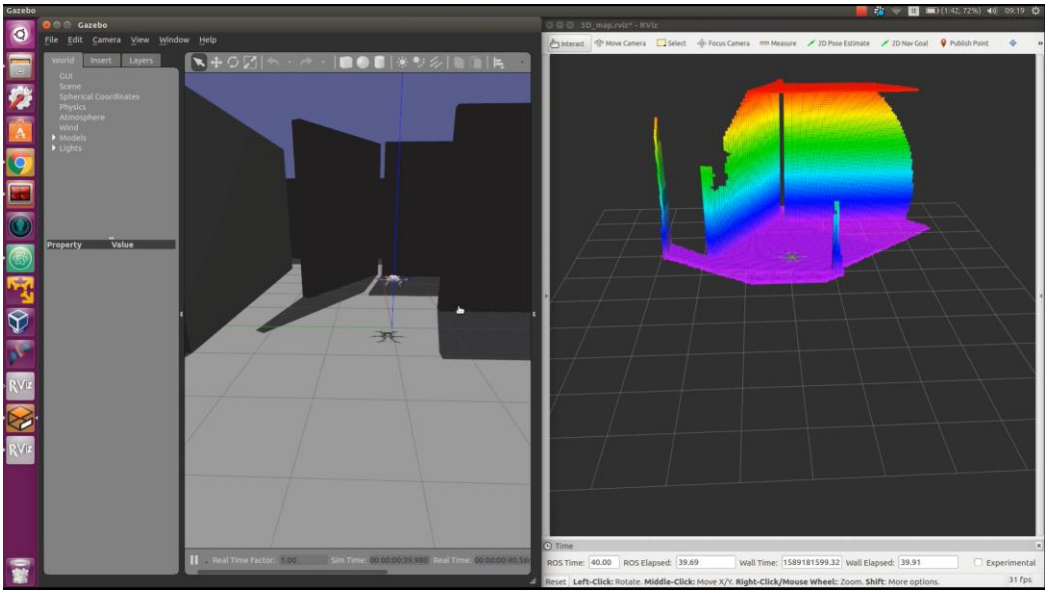


Figure 24 A screenshot from AirMPL-Simulator while a drone is perceiving the surrounding environment.

3.6.2 Improvements planned

The AirMPL-Simulator will be improved to allow the generation of virtual maps based on simulated data from the UGVs and UAVs on-board sensors. Further, the simulator will be endowed with controller algorithms so that the simulated autonomous agents will be able to move autonomously among obstacles. Also, a software interface between OMPL, AirMPL-Simulator and MATLAB will be developed.

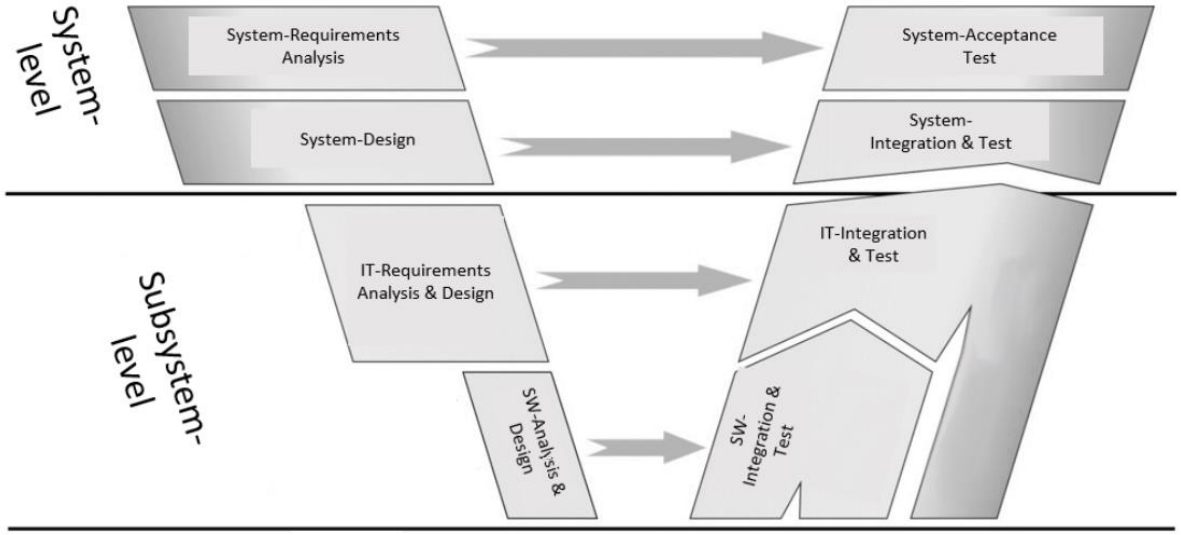


Figure 25 V-Cycle coverage of the AirMPL-Simulator.

3.6.3 Functional and non-functional requirements

UC5-DTC-41	The simulator shall simulate the path planner for performance evaluation.

3.6.4 Specific standard and/or regulation requirements

There are no specific standard and/or regulation requirements.

3.6.5 Specific installation and/or functional requirements

The minimum installation requirements are:

- Ubuntu Xenial (16.04) or Bionic (18.04).
- ROS Kinetic Kame or Melodic Morenia.
- Gazebo 7 or 9 distros.

3.7 SAGE Verification Suite (UNISS)

3.7.1 Tool Description

The SAGE Verification Suite is composed of several tools devoted to different verification tasks:

1. ReqV: Requirements Consistency Checking;
2. ReqT: Automatic Test Pattern Generation;
3. NeVer: Neural Network Verifier;
4. RLVer: Reinforcement Learning Policy Verification.

Considering the V model in Figure 1, the SAGE Verification Suite covers both sides of the System Level:

3.7.2 Improvements planned

UNISS will implement the following improvements:

1. ReqV: extension to more expressive logics and improvement of natural language processing capabilities. Actually, the system is able to take as input specification expressed as Property Specification Patterns encoded as Linear Temporal Logic extended with numerical constraints. During the project, ReqV capabilities will be extended to support more expressive logics, such as Signal Temporal Logic. This extension will allow to formulate technical specifications that can be translated in a logical language that is expressive enough in the context of the requirements listed above;
2. ReqT: extension to more expressive logics. Actually, the system is able to take input specification expressed as Property Specification Patterns encoded as Linear Temporal Logic. During the project, we will extend its capabilities in order to be able to deal with numerical constraints as a first step, and then ReqT will be extended to support more expressive logics, such as Signal Temporal Logic. Because ReqT approach is specification-based, in order to generate test suites, its input language must be aligned with the one used by ReqV;
3. Improve performance for MLPs and provide verification algorithms for NNs. This improvement is necessary for the formal verification of the algorithms developed in the context of what required by UC5-DEM10-FNC-07;
4. Provide verification algorithms for CTMCs, in order to cope with scalability issues of the current approach considering that the usage of a RL approach could be evaluated in the context of what required in UC5-DEM10-OPR-001/2.

3.7.3 Functional and non-functional requirements

UC5-DEM10-DTC-42	SAGE-VS should support the software criticality category defined for UAV standard (e.g., DO178C)
UC5-DEM10-DTC-43	SAGE-VS can be used to manage critical situations (task interferences, scheduling issues, fault of failure conditions)

UC5-DEM10-DTC-44	SAGE-VS can be used to verify the solutions for the management of critical situations, with improved situation awareness
UC5-DEM10-DTC-45	NeVer will support the design and the verification of AI algorithms used to detect and identify parasite animals and to classify leaf diseases using imaging sensor data

3.7.4 Specific standard and/or regulation requirements

NA

3.7.5 Specific installation and/or functional requirements

Linux OS.

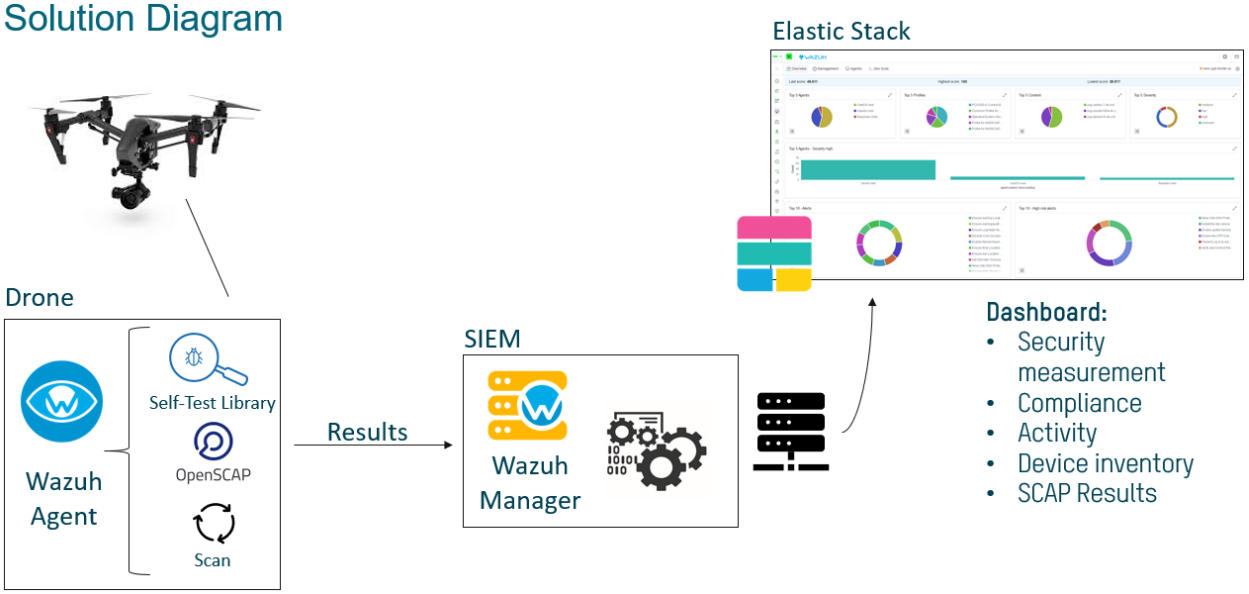
3.8 IoT Environment Extra-functional requirements validation and monitoring toolchain (IKERLAN)

3.8.1 Tool Description

The proposed toolchain will be a system start-up and runtime test and monitorization tool, with an on-board test system and a remote monitorization platform. Looking from a Safety operation perspective, the on-board test system will perform different start-up analysis and runtime tests of the correct operation, checking for the presence of faults and reporting them to the monitorization tool.

The system will be composed of two main elements. The first one will consist on a self-test checking tool, boarded on the drone device. That tool will perform different system checks when the main application starts and will also perform runtime diagnostics during normal operation. The different checks will identify if the system is running as expected or a fault is detected.

Solution Diagram



The second element will be a system monitorization element developed in WP5 and used in conjunction with the self-test checking tool. This part of the system will be the responsible of reporting the faults as soon as are detected to the final user. With that purpose, the tool will send the information from the device to an intermediate manager, and it will generate analysis reports for the final user.

3.8.2 Improvements planned

The integration of this toolchain in the drone device will improve the detection times of the user to identify any unexpected fault on the system at startup and runtime. Instead of the user looking for the faults, the

system itself will report the faults as soon as are detected, improving the detection and reaction time. That way, the overall reliability of the system will also be enhanced.

As the self-test are often required by different regulations (i.e. IEC 61508, ISO 26262), this will allow the development of safety functions in the device and reduce the effort of developing the safety functions compliant with the regulations.

The coverage of the V-Cycle of Figure 1 by the use of the proposed toolchain will be the one shown in Figure 26.

3.8.3 Functional and non-functional requirements

UC2-DTC-46	The system shall be monitorised in order to detect unexpected events

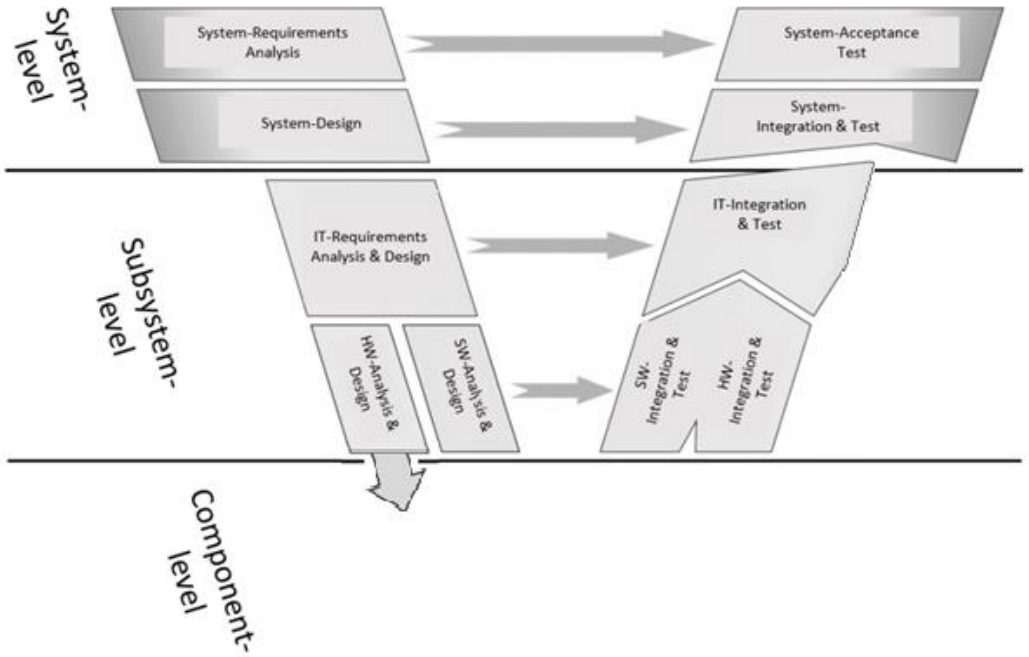


Figure 26 V-Cycle coverage of extra-functional validation toolchain

3.8.4 Specific standard and/or regulation requirements

The toolchain itself doesn't have direct standard or regulation requirements. However, it is interesting to note that the toolchain is useful to cover the self-test requirements for safety systems required by IEC 61508, ISO 26262, and any other regulation that requires system self-tests.

More precisely, it covers the requirements of IEC 61508-7 A.3.1, A.3.2 and A.3.3 (Self-test by software/hardware), and ISO 26262-5 D.2.3.1 (detection of failures in processing unit by software).

3.8.5 Specific installation and/or functional requirements

The drone device should allow the development and integration of the Self-Test Libraries and the scan and reporting tool. It should have an SDK for the development of the solution.

3.9 Event based IoT Environment validation methodology and toolchain (IKERLAN)

3.9.1 Tool Description

In the Internet of Things (IoT) vision, everyday objects have evolved into the so-called cyber-physical systems (CPS). These systems' use and deployment arise in industry giving place to Industry 4.0 or

Industrial IoT (IIoT). This kind of IIoT environments are distributed and asynchronous, where communication an interchange of messages is based on events (usually the communication approach used in these systems corresponds to a publish/subscribe paradigm).

While IIoT architectures have advantages like scalability and flexibility, they also face interoperability challenges among all participant agents in the network. Indeed, consistency problems raise when message content and its categorization get attenuated. These issues can also lead to information loss or complex processing requirements. This scenario does not fall that far from the environment presented in **COMP4DRONES**, where many and different drones can be working/collaborating together and receiving and sending messages.

The proposed solution is based on an existing tool called AsyncAPI Toolbox. This toolbox is a model-based proposal to design and develop this kind of architectures efficiently. In our solution we want to optimize the existing toolbox and extend it with a testing module that would facilitate the verification and validation of the communication consistency through-out the entire life cycle of the system. We want to foresee and avoid system malfunctions that can be the result of consistency failures among others, and have a system that has been extensively tested against common and unexpected failures. This way, as a result we could ensure a robust communication among CPSs.

3.9.2 Improvements planned

The proposed solution would decrease the detection time of consistency problems and facilitate the implementation of communication components. It would accelerate the integration of different elements along the system and fasten the verification tasks.

Considering the left side of the V-model, it would facilitate the modelling and documentation tasks. It would be the “single point of trust” for the representation of the communication interfaces that are used for the verification tasks.

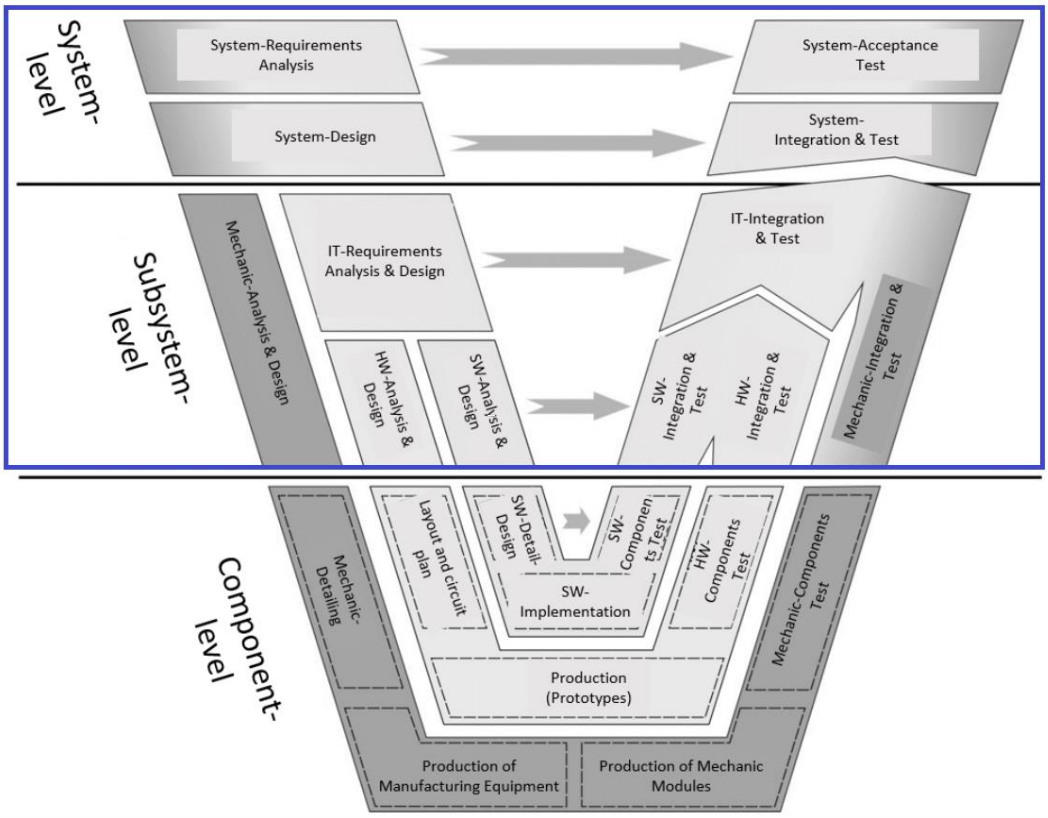


Figure 27 V-Cycle coverage of extra-functional validation toolchain

3.9.3 Functional and non-functional requirements

UC2-DTC-47	AsyncAPI toolbox shall allow to define the specification of the asynchronous communication protocol API (obtaining a single point of trust).
UC2-DTC-48	AsyncAPI toolbox shall allow to generate documentation related to the specification of the asynchronous communication protocol API.
UC2-DTC-49	AsyncAPI toolbox shall allow to generate source code related to the specification of the asynchronous communication protocol API.
UC2-DTC-50	AsyncAPI toolbox shall allow to generate tests to detect communication consistency failures.

3.9.4 Specific standard and/or regulation requirements

There are no specific standard and/or regulation requirements. The toolbox will offer support for the MQTT protocol.

3.9.5 Specific installation and/or functional requirements

The framework will be delivered as an Eclipse plug-in.

3.10 PhiSim (SHERPA)

3.10.1 Tool Description

The purpose of this tool will be to provide a simulation platform for drones. The simulation platform will be based on multiple open source software used in robotics. Priority would be given to the software that lets us build a modular system in the form of different components and provides a communication mechanism between each component. The software should also make the simulation of drone and sensor-related physics easy and simple.

At the end, we will obtain a simulation platform that would provide:

1. The possibility to model sensors (lidar and camera) in simulation
2. Modelling of drone aerodynamics (propellers, motors) as well as wind
3. Availability of different environments (buildings, forests)
4. Possibility to integrate new control and path planning algorithms with minimal efforts

This simulation platform will serve as a test bench for evaluating different algorithms developed in the work package 4.

The V-cycle coverage of PhiSim would be as shown in Figure 28.

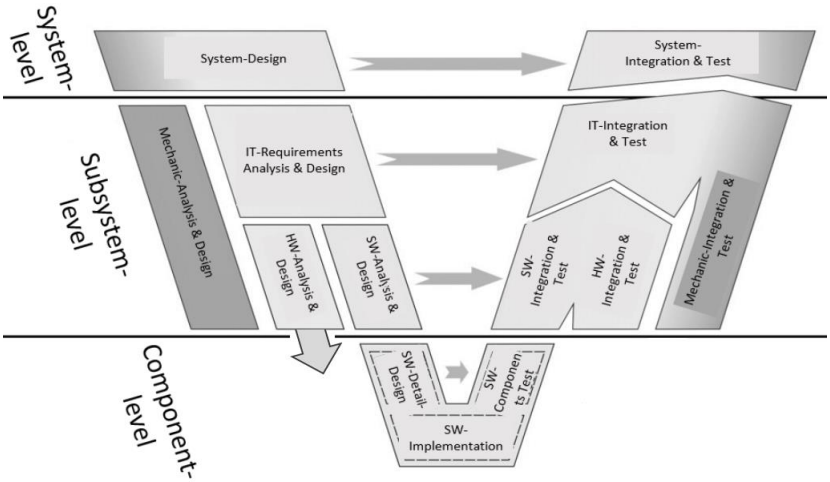


Figure 28 V-cycle coverage of PhiSim

3.10.2 Improvements planned

The following set of improvements are planned as part of this development:

1. Making the system customizable. Testing different controllers would be possible with minimal modifications. Efforts would be made to make the system modular. At the end of the project, it would be possible to test different algorithms (developed by **COMP4DRONES** partners) in the simulation platform with minimal efforts.
2. Bridging the gap between simulation and reality. This involves taking into account dynamic as well as thrust and wind parameters in order to have a realistic simulation.
3. Linking the simulation platform with Simulink.

3.10.3 Functional and non-functional requirements

UC3-DTC-51	The platform shall provide a possibility to simulate drones. To make the simulation environment close to reality, the parameters related to drones (aerodynamics, propulsion, wind etc.) shall be modifiable.
UC3-DTC-52	The platform shall simulate the sensors (Lidars, Camera) and the surrounding environment (buildings, trees etc. ...)
UC3-DTC-53	The platform shall allow to test different controllers and navigation algorithms with minimal efforts

3.10.4 Specific standard and/or regulation requirements

- Recommended OS: Ubuntu 16.04,
- Other Software: Open-source simulation platforms, Simulink,
- Programming Language: C++.

3.11 Path Management validation (ANYWI)

3.11.1 Tool Description

The purpose of this tool is to ensure communication reliability by exercising the path management facilities (WP5-12-ANYWI) of the multi-link communications subsystem. The tool shall simulate the availability and link state of various communications links using the API developed as part of WP5-13-ANYWI component and validate the path management's output against expectations.

The tool shall be able to record actual link state meta-information in a test environment and replay it in simulation.

3.11.2 Improvements planned

The integration of this tool into the toolchain will permit developers of link drivers and firmware to test their link's effects on the reliability management of the multi-link communications subsystem.

3.11.3 Functional and non-functional requirements

UC4-DTC-54	The toolkit shall test and validate that available communication links via different operators (in case of mobile connections) or access technologies are discovered and made available to the path management system of the multipath communication system on the drone.

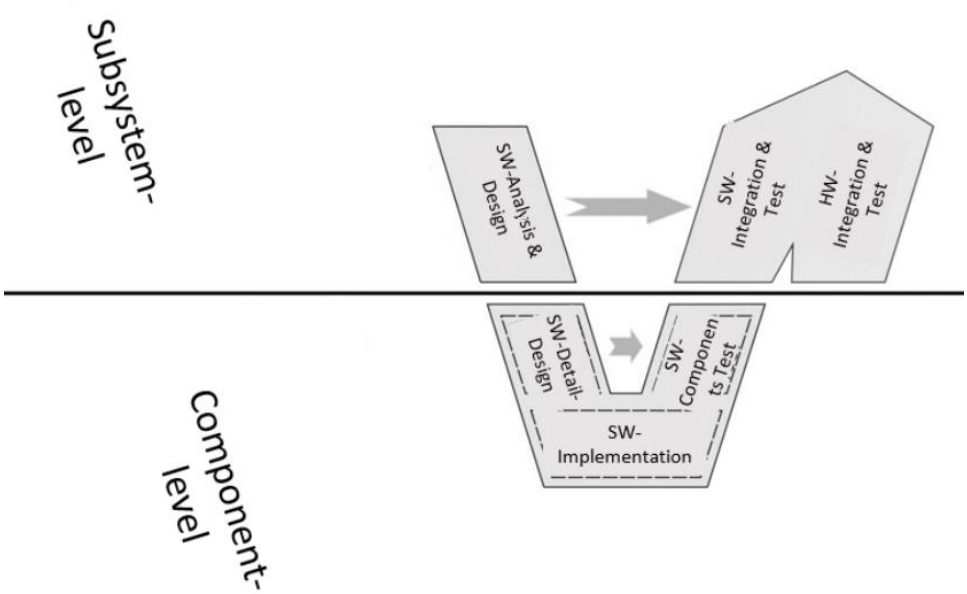


Figure 29 V-cycle coverage of Path Management validation.

3.11.4 Specific standard and/or regulation requirements

Linux OS

3.12 Link State validation (ANYWI)

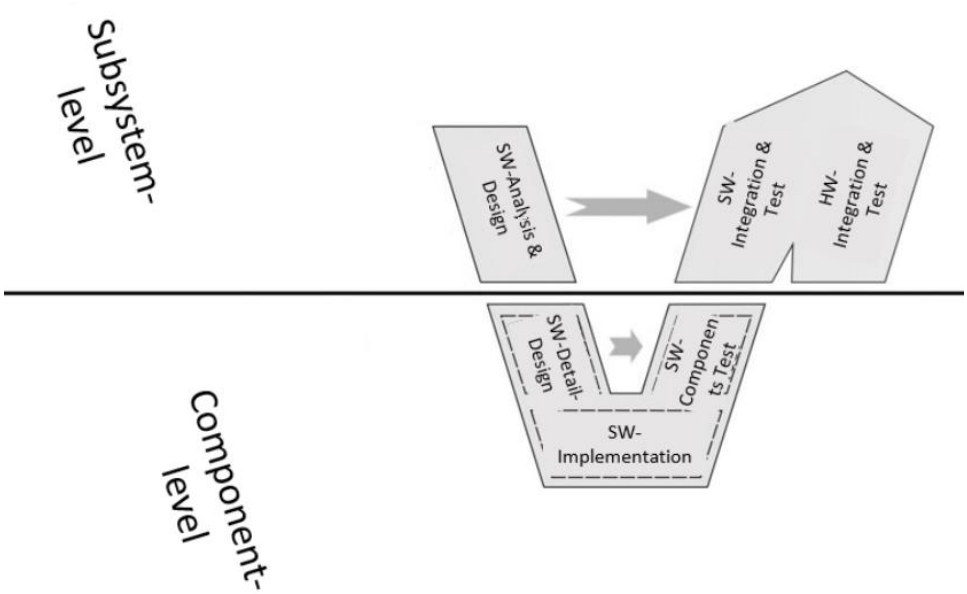


Figure 30 V-cycle coverage of Link State validation.

3.12.1 Tool Description

The tool shall provide facilities for validating link state meta-information as provided by communication link drivers and firmware.

3.12.2 Improvements planned

For the correct functioning of the link reliability API (part of WP5-13-ANYWI component), driver and firmware developers must be able to verify that the link state meta-information they produce can be consumed by the API, and the API produces valid results equivalent to the ingested data.

3.12.3 Functional and non-functional requirements

UC4-DTC-55	The toolkit shall validate the collection of metadata from data connections between drone and ground.

3.12.4 Specific standard and/or regulation requirements

Linux OS

4 System Analysis and Optimization tools

In this section, the tools which have as main objective the validation and/or verification of the design process, are described.

4.1 Security Analysis Tool (AIT)

4.1.1 Tool Description

AIT developed a threat modeling tool which is integrated into Enterprise Architect in order to support a continuous model-based engineering and connect the system model used for security engineering with the system model used for security engineering.

ThreatGet is a threat analysis and risk management tool, specifically developed for IoT and embedded systems with a potential impact on the physical world. Figure 31 shows the basic architecture of the tool. The web-based Backend is used to maintain the knowledge base about threats and mitigation measures and to conduct the analyses. Multiple user can work on the same backend and cooperate regarding threats.

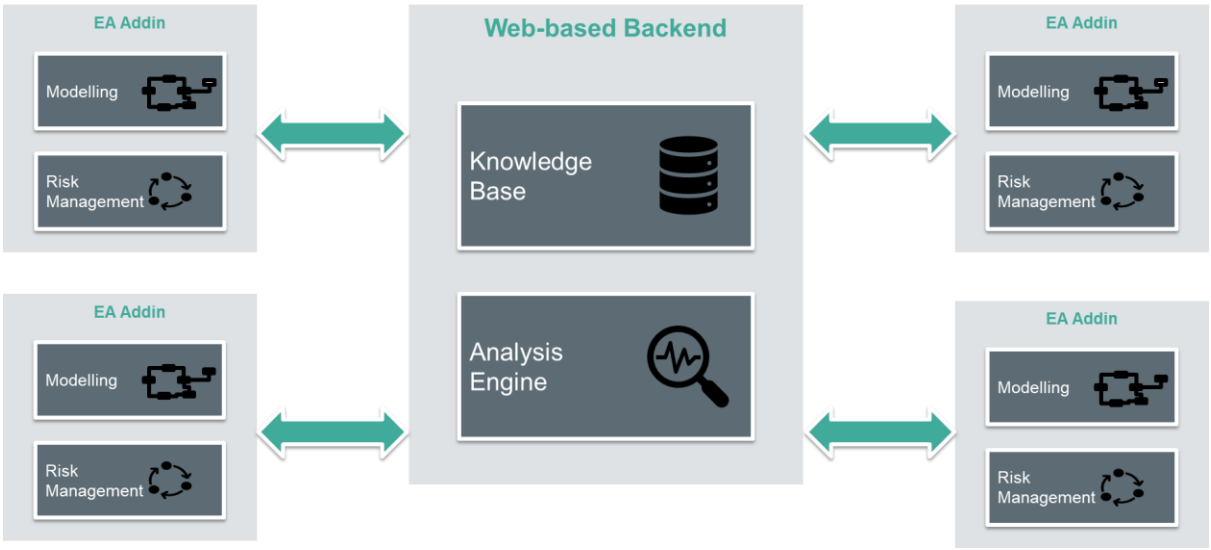


Figure 31 ThreatGet System Architecture

In the backend a library of domain specific elements and security properties is maintained. This can be used to model a system from a supported domain with all relevant security properties. After the analysis, the identified threats can be evaluated, and risk treatment decisions are documented.

Figure 32 shows the ThreatGet modeling UI with a simple example. Here the system is modelled, and security properties are assigned. The analysis results in more less threats. Treatment decisions are documented and assigned to the elements, resulting in an acceptable level of risks.

Currently the tool I usable in the design phases and the modelled threats cover mainly system/sub-system level.

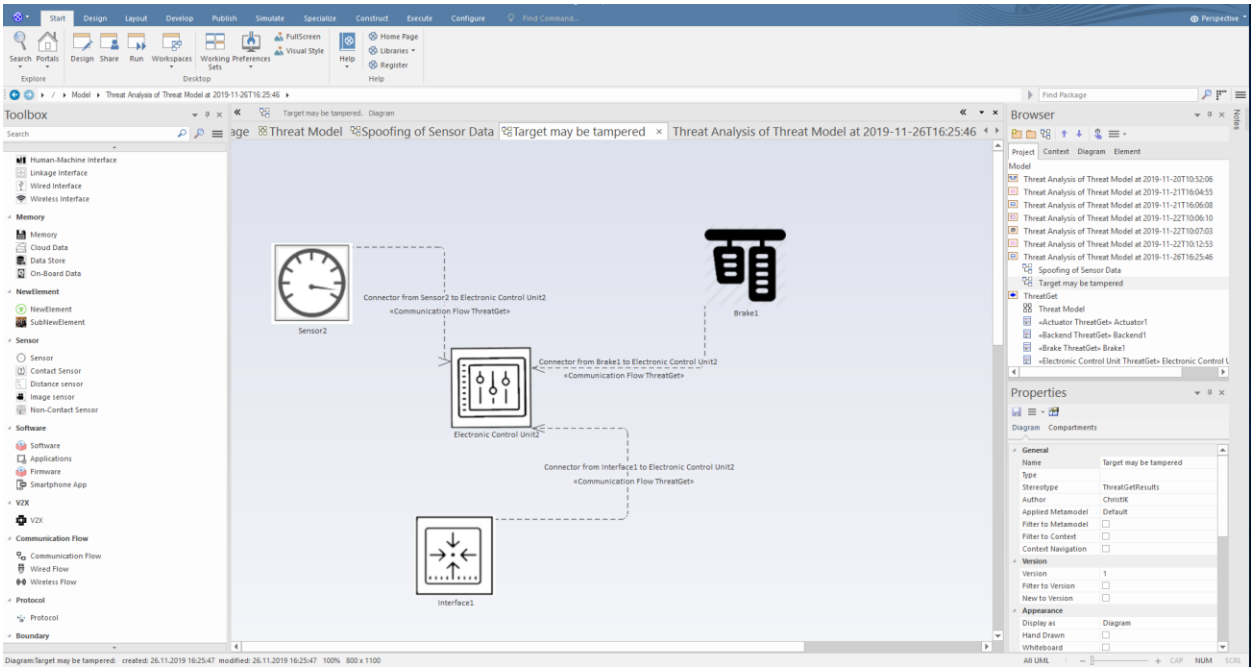


Figure 32 ThreatGet Modeling UI.

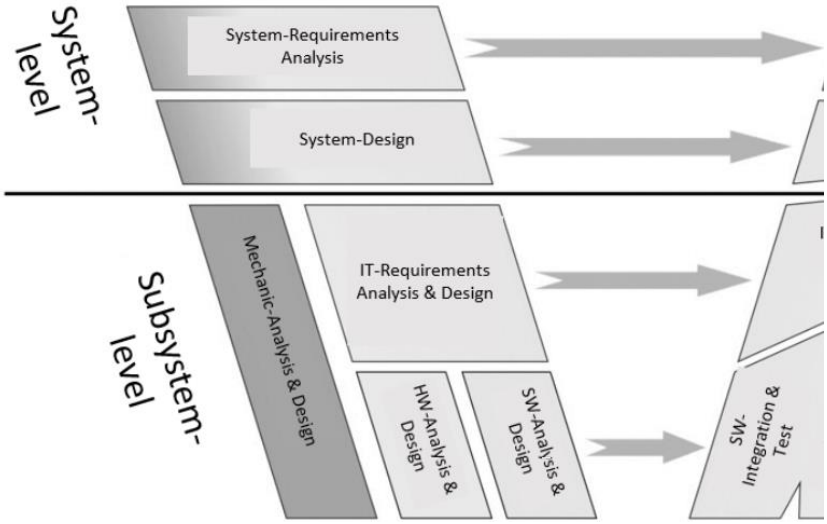


Figure 33 Intended coverage of the AIT Security Analysis Tool.

4.1.2 Improvements planned

We plan two major improvements:

- The current list of threats stored in the system is either generic or developed for automotive. This needs to be extended to make the tool applicable for drone systems.
- In order to support a modular engineering approach a component level threat diagram needs to be integrated into a system-level threat diagram with combination of threats and consideration of security guarantees and assumptions

4.1.3 Functional and non-functional requirements

UC5-DEM9-DTC-56	Safety-security co-analysis
UC5-DEM9-DTC-57	The communication between land-bound sensors and drones shall be trusted

UC5-DEM9-DTC-58	The communication between drones and base station shall be trusted

4.1.4 Specific standard and/or regulation requirements

The diagram is based on Dataflow Diagrams

4.1.5 Specific installation and/or functional requirements

The tool will be developed as an Enterprise Architect plug-in.

4.2 Safety Analysis Tool (BUT)

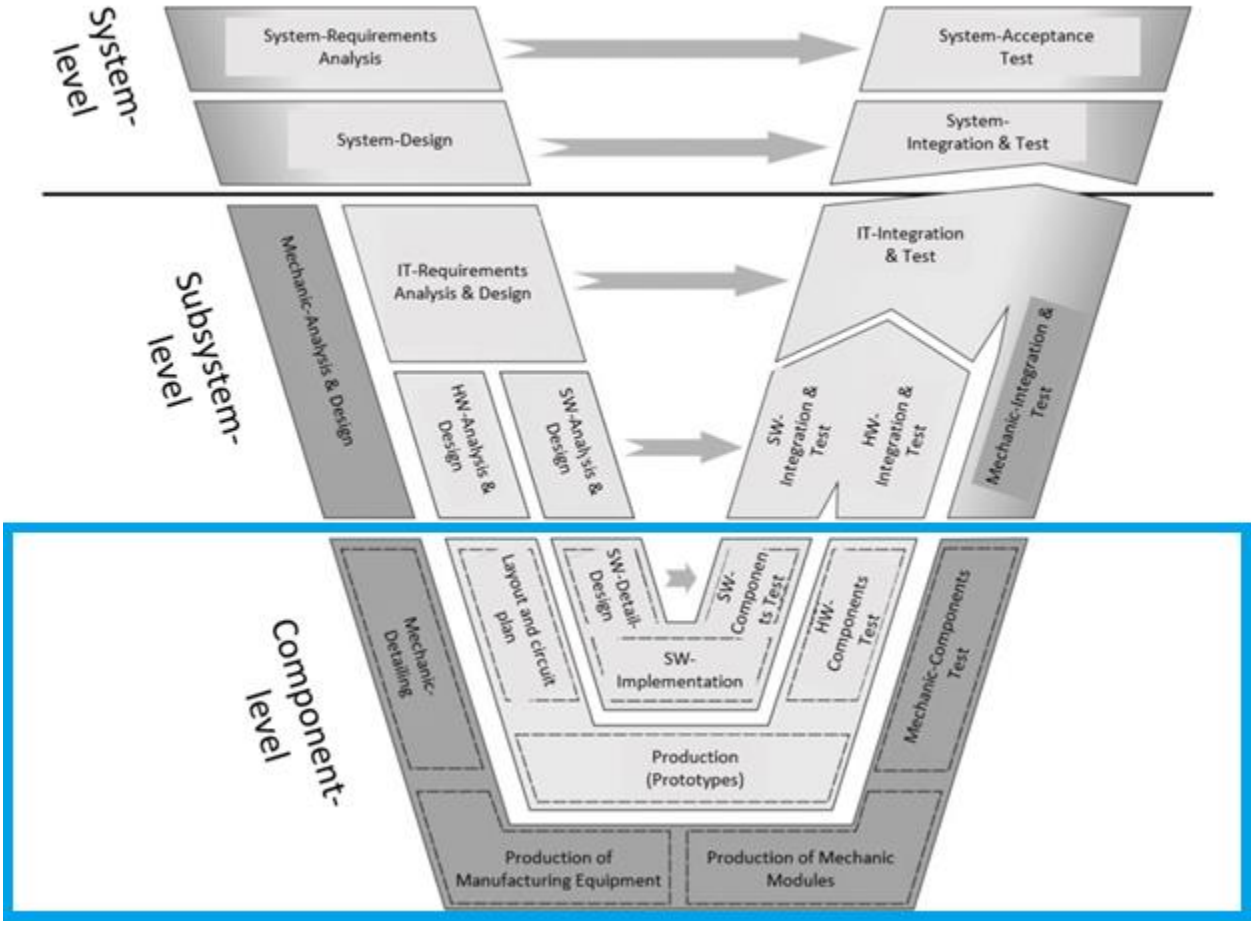


Figure 34 V-Cycle coverage of Safety Analysis Tool.

4.2.1 Tool description

The tool support provided to **COMP4DRONES** is a collection of plugins developed at FIT BUT for the Facebook Infer and Frama-C frameworks whose aim is to perform static analysis of pieces of software that are safety or otherwise critical. The plugins specifically aim at concurrency-related and performance-related features, but they are accompanied with other analyses available in Facebook Infer and Frama-C that aim at other kinds of code defects (arithmetic flaws, memory safety, etc.).

4.2.2 Improvements planned

The mentioned tool support will be adopted and evaluated on a piece of software from FIT BUT (e.g. HDR processing software) and offered to **COMP4DRONES** community as an Open Source Software.

4.2.3 Functional and non-functional requirements

UC4-DTC-59	Verification tool for drone software which perform static analysis of pieces of software that are safety or otherwise critical

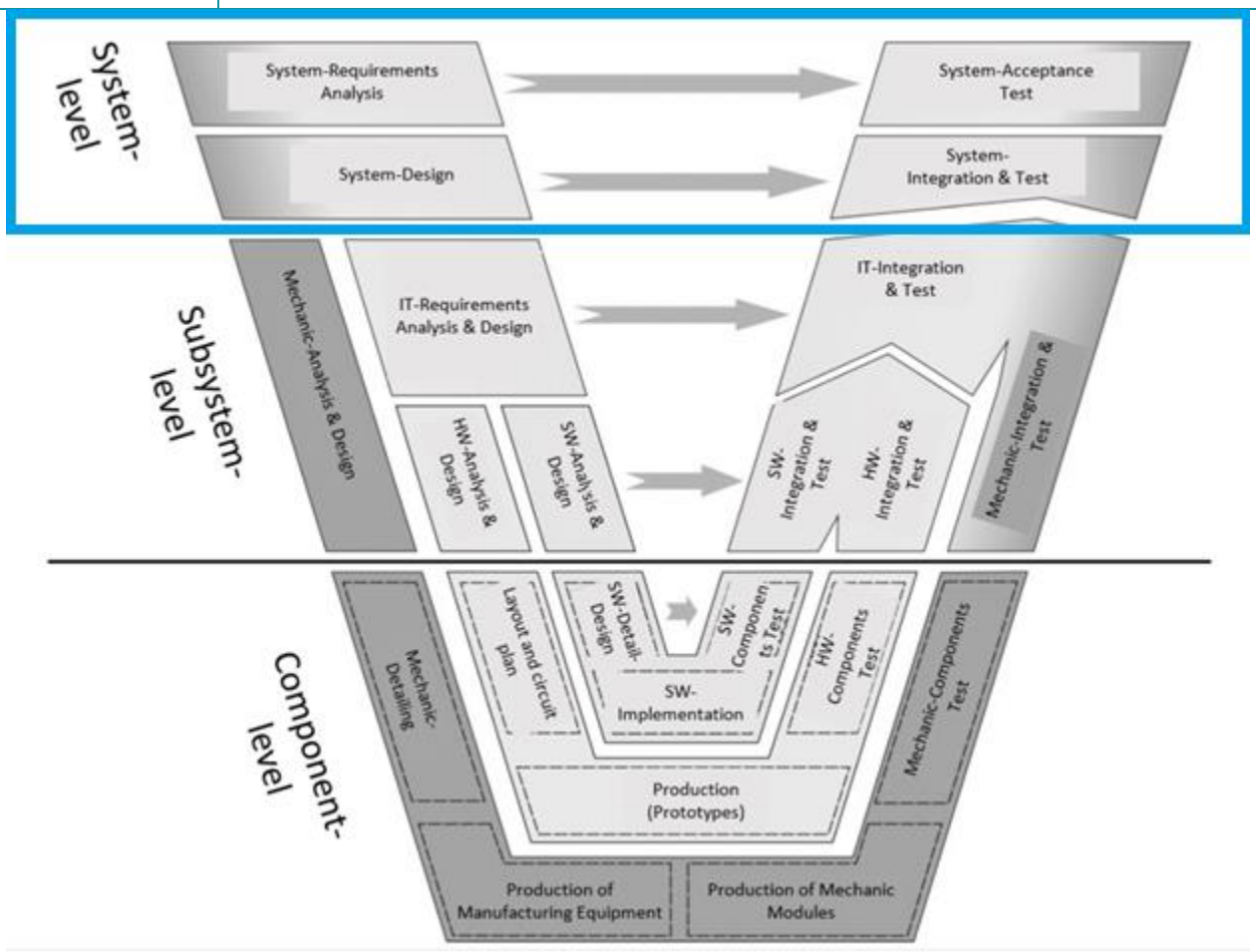


Figure 35 V-Cycle coverage of Big Data analytics tool.

4.3 Big Data analytics tool (BUT)

4.3.1 Tool description

The custom tool is merely a Big Data analytics system for object detection and possibly categorization as well as recognition. The tool will consist of object detection, extraction of objects from video/images, storage of the images and processing of a data base of the objects, (semi) automatic annotation of the object data base, and analytics/machine learning sub-tools for processing of the data base (AdaBoost/WaldBoost and/or Random Forests and/or CNN based tools, etc.).

4.3.2 Improvements planned

In **COMP4DRONES**, BUT is developing this tool from scratch. We will consider third party tools/libraries for big data analysis could be incorporated.

4.3.3 Functional and non-functional requirements

UC4-DTC-60	The tool shall provide data analysis capabilities. It shall be possible to detect different objects from video/images.

UC4-DTC-61	The Big Data Analysis tool should unify and simplify the processing of acquired data.

4.4 Simcenter Amesim (Siemens)

4.4.1 Tool Description

Siemens' Simcenter Amesim is a software tool dedicated to modelling and simulation of dynamic and multi-physics systems.

In the tool's environment, systems are modelled connecting components available in the libraries, which cover several physical (fluids, mechanical, electrical…) and application (aerospace, automotive, gas turbines…) domains.

Each component is described by tabulated data and/or nonlinear time-dependent analytical equations. More than one modelling option can be proposed by the same component (i.e. tabulated approach or analytical equation), and the choice is made by selecting the preferred sub-model. This mixed-fidelity approach provides the benefit of scalable modelling strategy, where the model accuracy can evolve along with the design cycles as design decisions are made and more information of the product becomes available. Sub-models are coded with the programming language C.

The components interface is realized through ports that allow the flow of physical variables. The definition of the interfaces is based on the bond graph theory, which allows a common representation of dynamic systems regardless their physical domain. This approach ensure consistency in the conservation of energy, conservation of mass and units of measurements when connecting different components.

Simcenter Amesim is equipped with a solver which automatically adapts the time step and selects the best algorithm for the resolution of the systems of equations. This allows faster simulations, and at the same time, it let the user focus on the modelling aspects instead of having to take care of the choice of the solver parameters. The tool also provides the possibility to use fixed step solvers, which is required for co-simulations with other software tools or Hardware-in-the-loop, Software-in-the-loop, and Real-Time activities.

Figure 36 Screenshot of Simcenter Amesim GUI

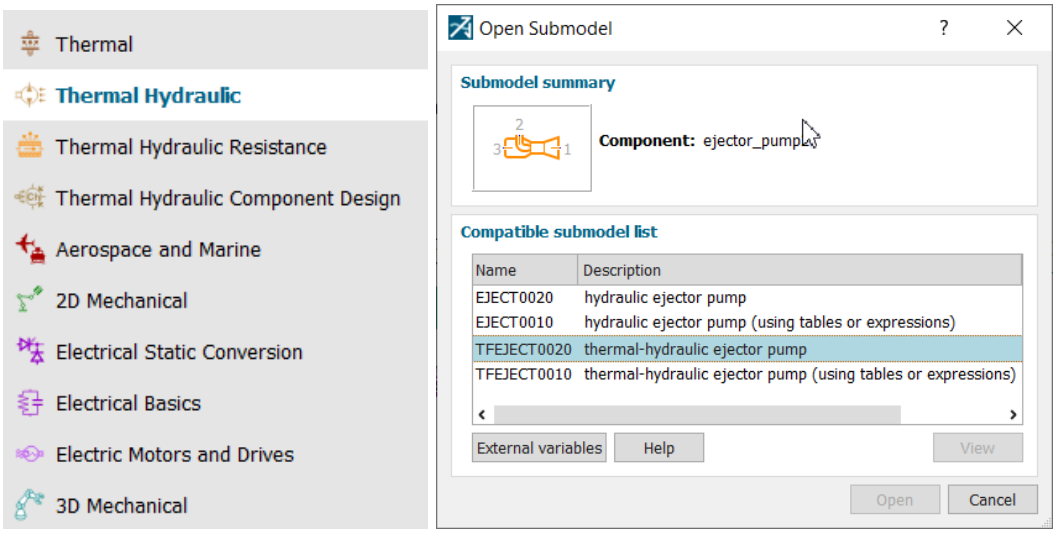


Figure 37 On the left, portion of the library tree. On the right, list of sub-model associated to the same component.

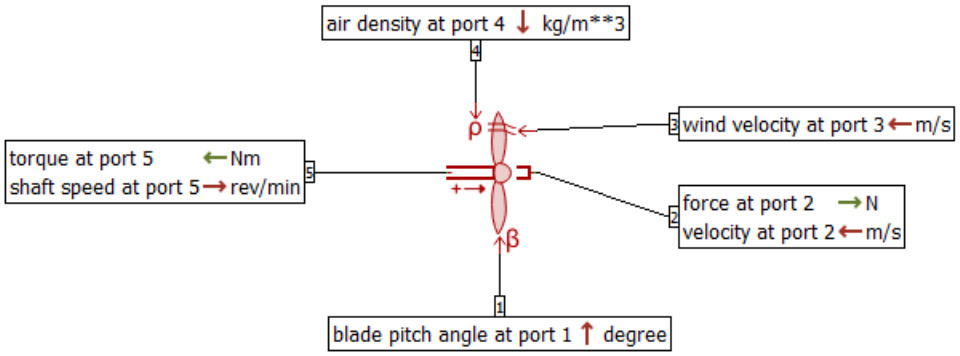


Figure 38 Example of component interface. The propeller component uses signals, linear and rotational mechanical ports.

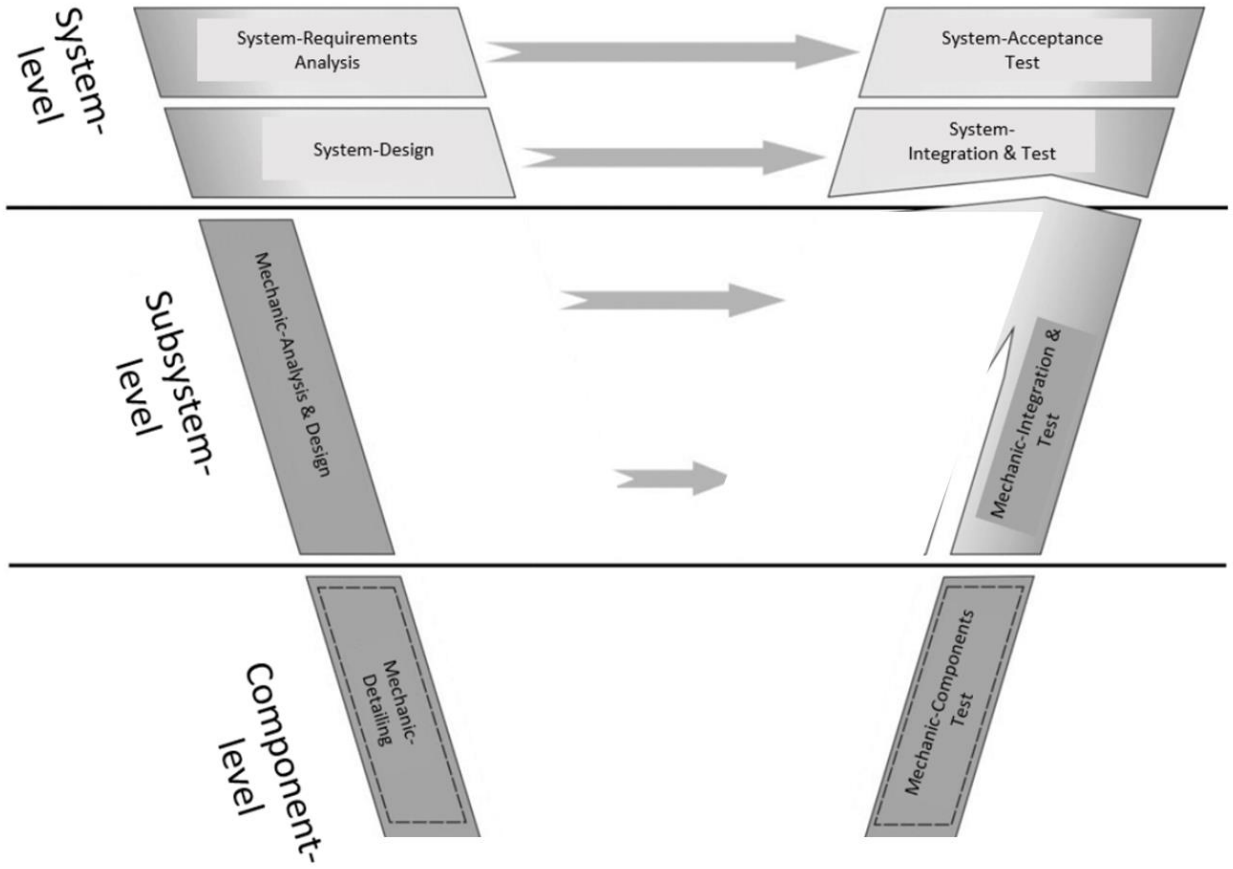
Simcenter Amesim is by nature an integration platform, with dedicated interfaces to other software tools. Siemens is part of the consortium defining and developing the Functional Mock-Up Interface (FMI) standard. This is a protocol that defines a container and an interface to exchange dynamic models using a combination of XML files, binaries and C code zipped into a single file. FMI can be considered as an enabler for scalable and tool neutral integration of simulation models from different technical disciplines, developed by different actors. Amesim openness is an asset for this research project, and it will be used to couple drone's system simulation model to other tools modelling complementary aspects of drone's design, analysis and optimization. For example, it can be coupled with Simcenter Prescan, another software tool of Simcenter portfolio, which provides sensors and environment modelling capabilities for autonomous vehicle simulation.

There are four key benefits that Simcenter Amesim brings to drones design, analysis and optimization:

1. Performance analysis. The tool allows to simulate the drone's performance at any time for different missions and conditions. This help determining performance requirements, verify design choices and analyse eventual extensions of drones' operability,
2. Virtual systems integration. It is possible to achieve virtual systems integration from the early phases of the design cycle. By integrating different systems and physical domains, it is easier to discover unexpected interactions before physical prototypes are available. The tool's openness and integration capabilities also support collaborative engineering,

3. Scalability. Simcenter Amesim supports the product lifecycle, from early design to entry into service and operations. The tool scalable approach allows to increase model accuracy as the design matures,
4. Modularity. Thanks to its libraries and components, the tool allows to easily build and evaluate different architectures and variants of a given product. It makes it possible to adapt quickly existing models to new architectures/variants and perform trade-offs to support design decisions.

Thanks to the capabilities aforementioned, Simcenter Amesim can provide an effective contribution in the phases of the V-cycle depicted below.



4.4.2 Improvements planned

The following list is non-exhaustive and is likely to be adapted as the project advances:

- Provide functionality to generate a mission profile (altitude and speed definition for different flight phases)
- Develop component for coaxial propeller performance
- Improve fidelity of aerodynamics sub model for UAV applications
- Improve (with dedicated developments or through methodologies) the integration with other tools for environment simulation, sensors simulation and flight simulators. The goal is to provide a comprehensive simulation framework to address autonomous drone simulation.

- Industrialize demonstrators to help users understand the software capabilities and provide a starting point for UAV system simulation analysis

4.4.3 Functional and non-functional requirements

UC3-DTC-62	The tool shall provide system simulation integration capabilities. It shall be possible to integrate different physical domains as well as integrate third party tools.
UC3-DTC-63	System modeling time shall be reduced by 20% with respect to current methods
UC3-DTC-64	The tool shall provide a modular approach to system modeling. The goal is to allow models reusability, minimizing the time needed to adapt existing models to simulate novel drone configurations. The adapting time shall be reduced by 20% with respect to current methods.

4.4.4 Specific standard and/or regulation requirements

There are no specific standard and/or regulation requirements.

4.4.5 Specific installation and/or functional requirements

The minimum installation requirements are listed below:

- Windows or Linux, Only 64-bit operating systems are supported
- Processor 64-bit Intel® or compatible processors (AMD) except Intel® Itanium® 1 and 2.
- RAM 4 GB. Disk space 18 GB.

4.5 MoSaRT (ENSMA)

4.5.1 Tool Description

Drones as safety-critical real-time systems need to be analysed at an early stage of the development life-cycle in order to check if all the timing requirements are met. Indeed, one of the main difficulties that the system designers face is to find the appropriate analysis tests helping to validate and/or to dimension properly their designs. MoSaRT (Modeling oriented Scheduling analysis of Real-Time systems) is a model-based framework for design and analyse real-time systems. It covers the modelling cycle and aims to provide an oracle to select the best feasibility tests, and the best tools offering these tests for single processor cases, multiprocessor cases, and distributed cases.
On the one hand, MoSaRT provides a graphical design language which is compliant with standard ones and which leads to get accurate design. One the other hand, MoSaRT also provides an analysis repository model playing the role of a storage of research studies enabling researchers to promote their works (e.g. analysis models, schedulability analysis, dimensioning tests), then to increase the applicability of the real-time scheduling analysis.

4.5.2 Improvements planned

Two possible enhancements. The first one is the adaptation of the modeling language to support drones specificities in term of temporal behaviour. The second enhancement will be related the enrichment of the repository by adding adequate contexts and model transformations toward external appropriate analysis tools.

The following figure highlights the V-cycle phases where MoSaRT can be used.

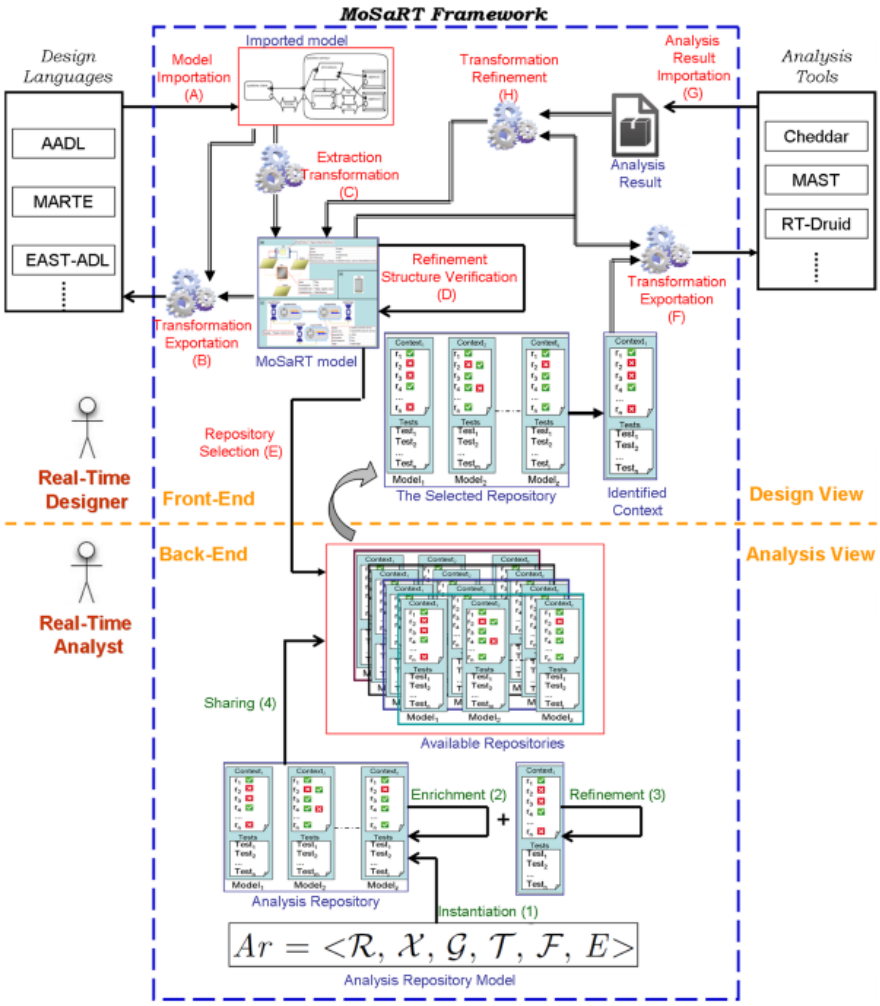


Figure 39 MoSaRT framework

4.5.3 Functional and non-functional requirements

UC3-DTC-65	MoSaRT shall reduce the design and analysis efforts
UC3-DTC-66	MoSaRT shall allow designers to check the schedulability of their models without being expert
UC3-DTC-67	MoSaRT shall allow to compare different analysis results providing by different analysis tools

4.5.4 Specific standard and/or regulation requirements

- The framework is based on its own meta-model but it supports UML/MARTE and AADL.

4.5.5 Specific installation and/or functional requirements

The framework will be delivered as an Eclipse plug-in.

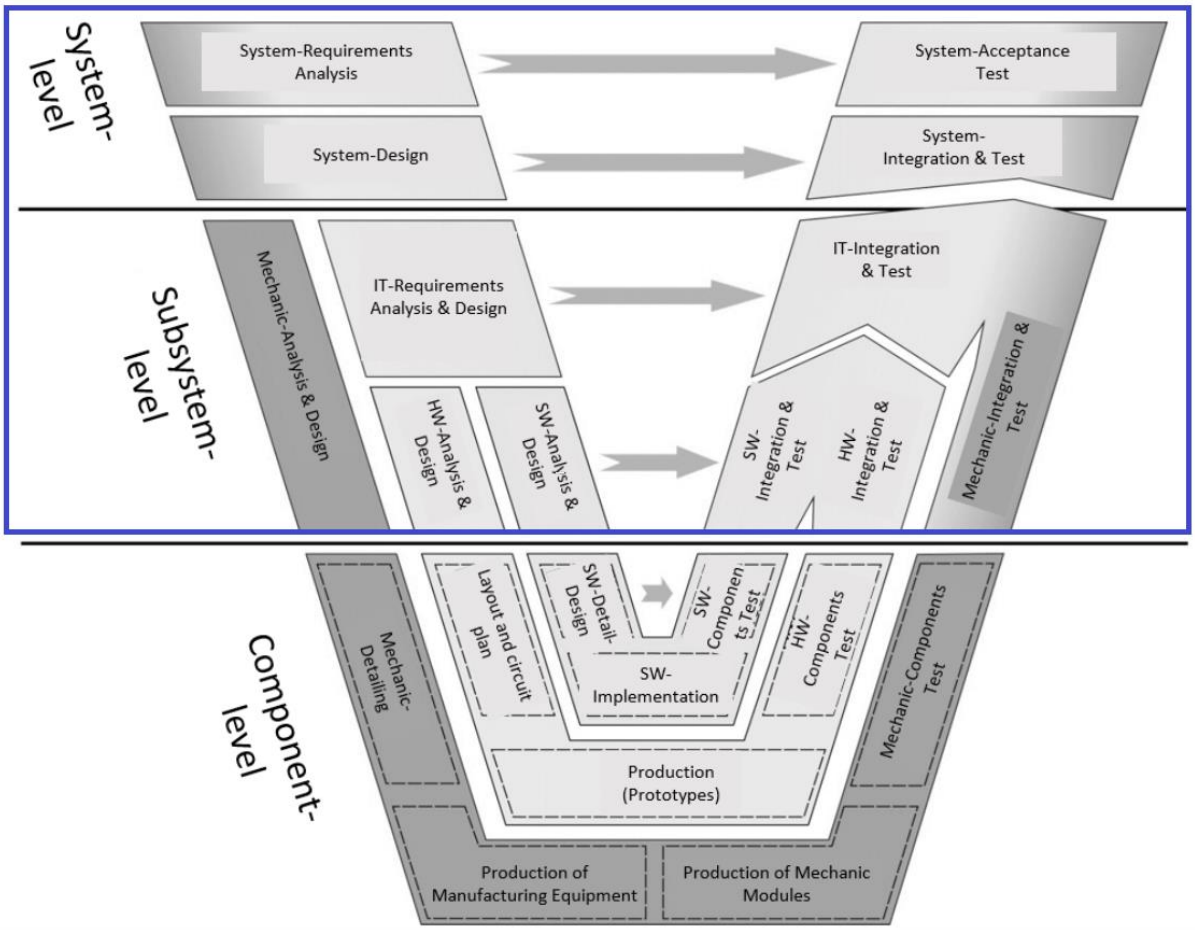


Figure 40 V-Cycle coverage of Mosart.

4.6 IPS MAF: Indoor Positioning System Model & Analysis Framework (ACORDE)

4.6.1 Tool Description

The Indoor Positioning System Model&Analysis Framework (IPS MAF) enables the description of the deployment of the anchor beacons and of a tag in a given infrastructure, the modelling of different possible behaviours of the anchors and the tag, and the analysis of its impact on the estimated position, and the analysis of the Dilution of Precision (DOP) on the infrastructure. Specifically, the high-level trilateration algorithms employed by the anchors for its automatic geo-positioning, and the positioning algorithm of the tag can be modelled. The model is serving for simulating the impact of sensor accuracies, of the anchor and tags algorithms, of the specific deployment and, at a high level, of the type of medium access protocols employed. Moreover, a specific I/O interface is provided to enable its exploitation on the customization of the IPS, either by ACORDE (provided as a custom IPS service) or by a final customer acquiring the IPS.

4.6.2 Improvements planned

In **COMP4DRONES**, ACORDE is developing the IPS MAF from scratch, on top of a basic development framework relying on Eclipse, SystemC and Octave. Eventually, other third-party tools/libraries for visualization and deployment description could be incorporated.

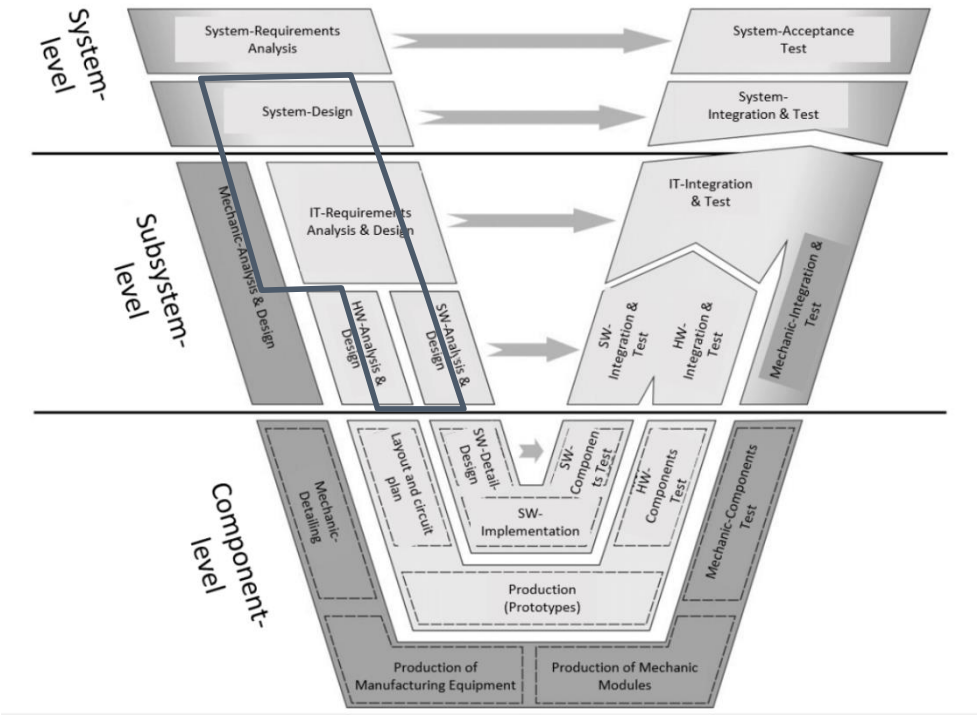


Figure 41 V-Cycle coverage of the IPS MAF.

4.6.3 Functional and non-functional requirements

UC2-DEM2-DTC-68	Integration on graphical environment for edition, building, and simulation of indoor positioning system models
UC2-DEM2-DTC-69	Integration of version control system on the IPS MAF
UC2-DEM2-DTC-70	Support modelling of based on standard language/procedures
UC2-DEM2-DTC-71	Support of simulable models, with integrated model of time
UC2-DEM2-DTC-72	Support high-level modelling (effects on latencies) of medium access protocols
UC2-DEM2-DTC-73	Interface for describing the deployment of the UWB anchors
UC2-DEM2-DTC-74	IPS model output which can be directly passed to 3rd party tools for user friendly graphical analysis, e.g. Matlab, Octave, Gtkwave.
UC2-DEM2-DTC-75	No license costs
UC2-DEM2-DTC-76	Scalable simulation time, allow fluent design of small models (e.g. minutes max. for less than 10 anchor models), few hours for a complex model (dozens anchors, with the highest level of detail on the medium access level, the complete flight simulation, including warm-up and normal run phases).

4.6.4 Specific standard and/or regulation requirements

The following standards are of specific interest for the environment:

- IEEE Std. 1666-2011 (SystemC Language)
- C++ coding standards
- ARM's Cortex Microcontroller Software Interface Standard (CMSIS)

4.6.5 Specific installation and/or functional requirements

The following installation requirements have been defined so far:

- Linux Ubuntu 18.04LTS or Windows 10.0 based environment.

- SystemC v2.2 (tested with MinGw64 v4.8.3 on windows)
- Eclipse CDT (or Polarsys v0.8)

4.7 SoSim (UNICAN)

4.7.1 Tool Description

SoSim will be an extension of VIPPE, an optimum simulation-based performance assessment technology for Design Space Exploration (vippe.unican.es), to simulate and estimate performance for complex CPSoS.

VIPPE is a simulation and performance analysis tool based on native simulation, a fast simulation technology enabling simulation and performance analysis of complex systems. VIPPE provides time, power and energy metrics. Breakdowns per platform elements (processors, buses, memories) and information on metrics like cache performance are provided to let the early analysis of performance bottlenecks.

The design steps of the V-Cycle covered by SoSim as part of the S3D Single-Source System Design Framework as already shown in Figure 18.

4.7.2 Improvements planned

VIPPE will be extended to the simulation and performance analysis of Cyber-Physical Systems of Systems (CPSoS) so that it is possible to simulate, analyse and explore different architectural mapping for complete missions in a short time. The new tool with these new features will be named SoSim.

4.7.3 Functional and non-functional requirements

UC2-DTC-77	SoSim will be able to simulate a complete drone mission. The simulation model will be automatically generated from the S3D model
UC2-DTC-78	SoSim will support performance estimation of non-functional metrics such as energy and delays
UC2-DTC-79	SoSim will accelerate performance analysis in more than 20%.

4.7.4 Specific standard and/or regulation requirements

The framework uses UML/MARTE as background meta-model. Nevertheless, this is made transparent to the user.

4.7.5 Specific installation and/or functional requirements

The framework is delivered as an Eclipse plug-in.

4.8 ROS1 & ROS2 infrastructure/dev-ops (ALM)

4.8.1 Tool Description

Providing a standard setup for the development of robotics using ROS1, ROS2 and MAVLink interfaces. Focus of this standard setup is on the interoperability between the two versions of ROS and external dependencies. Its role in the C4D Development V-Cycle is shown in Figure 42.

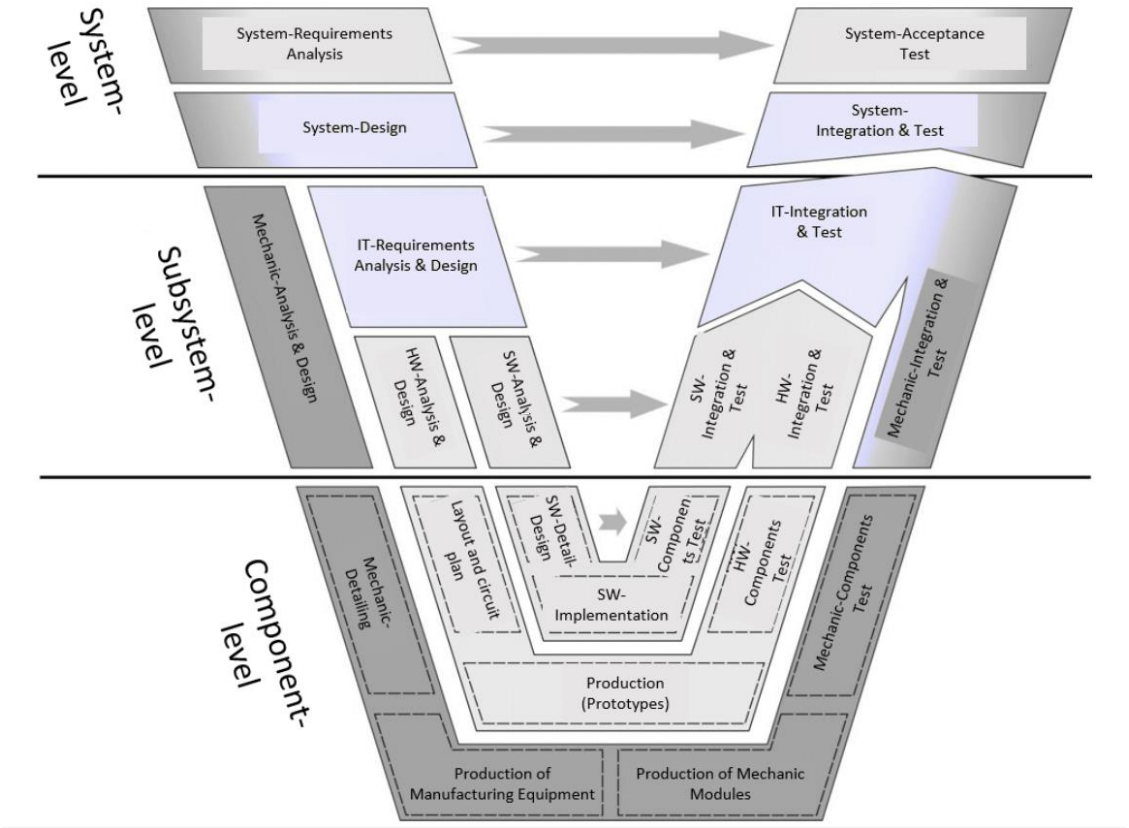


Figure 42 V-Cycle positioning.

4.8.2 Improvements planned

Migration from ROS1 to ROS2 is not entirely without effort, as there are several fundamental technology changes between the versions. This holds for the communication layer, but also for the development tools, library dependencies and documentation. Through a bridge node it's already possible to have a ROS2 environment connecting and sharing messages with a ROS1 environment.

4.8.3 Functional and non-functional requirements

There are only a few actual requirements from the UC related to this component. Most UCs and most specifically UC4 are working on a combination of ROS1 and ROS2, and interaction with MAVlink, other auto-pilots, etc.

From UC4 there is requirement: UC4-DEM2-DTC-04 (UC4-PRF-12) COTS standards for communication - ROS2, MAVlink, WIFI, etc.

UC4-DEM2-DTC-80	(UC4-PRF-12) COTS standards for communication: ROS2, MAVlink, WIFI, etc.

4.8.4 Specific standard and/or regulation requirements

No relevant regulations.

4.8.5 Specific installation and/or functional requirements

The component provides a standard template ROS1/ROS2 workspace, through a Catkin template.

4.9 Cloud-based simulation environment (ALM)

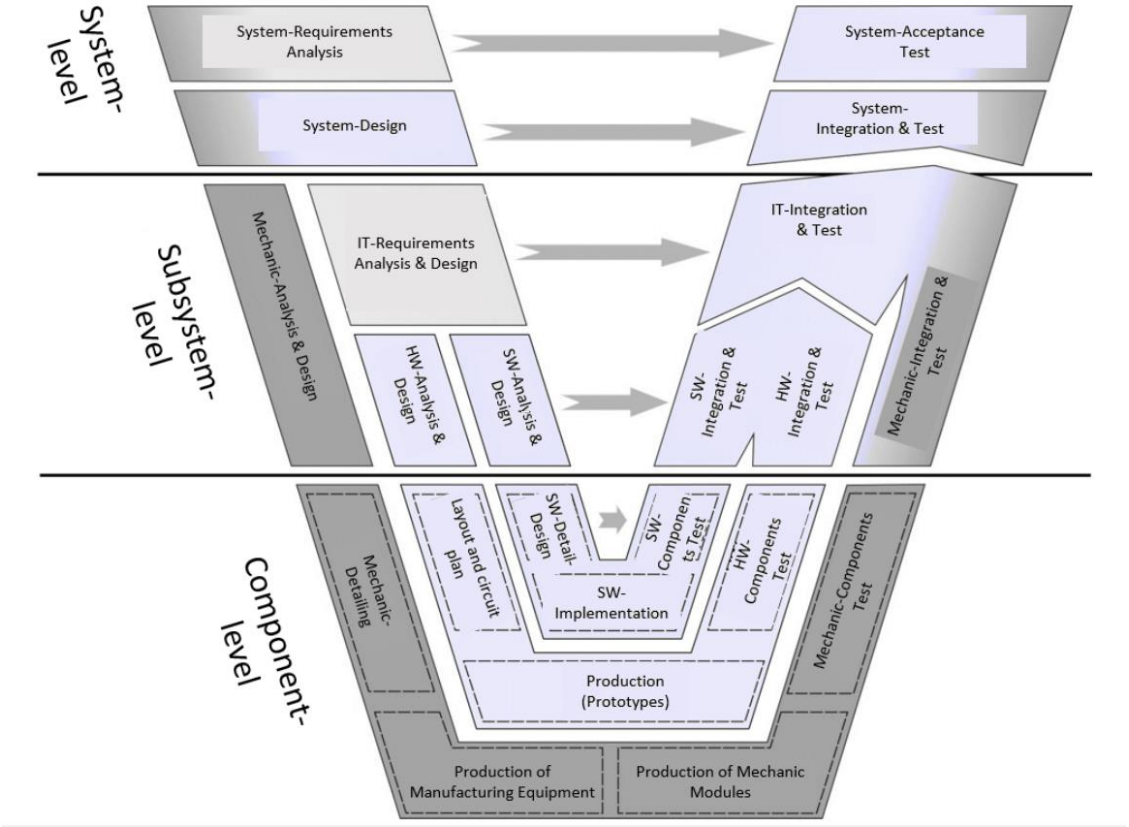


Figure 43 V-Cycle positioning.

4.9.1 Tool Description

Simulation of UC4Demo2, Single rover, multiple drones in an indoor environment, (e.g. smoke-simulation, point of interest, 3-D obstacles) Cloud hosting of the simulation for the project. Working towards hardware-in-the-loop as well. Explore AR/VR options. Its positioning in the C4D Development V-Cycle is shown in Figure 43

4.9.2 Improvements planned

Current state-of-the-art is Gazebo+Single-ROS-environment+locally hosted, We aim to introduce multiple ROS environments (=Multi-master), cloud-hosted, simulation (probably Gazebo, optionally others).

4.9.3 Functional and non-functional requirements

UC4-DEM2-DTC-81	(UC4-PRF-05) Common model creation: SLAM (SLAM Modelling and Simulation) (semantical rich modelling)
UC4-DEM2-DTC-82	(UC4-PRF-10) User Interfaces: High level mission control
UC4-DEM2-DTC-83	(UC4-PRF-11) User Interfaces: Live model visualization

4.9.4 Specific standard and/or regulation requirements

No relevant regulations.

4.9.5 Specific installation and/or functional requirements

Cloud hosting will probably be done on AWS infrastructure.

4.10 Stakeholder Acceptance Digital Test Bench (ALTRAN)

4.10.1 Tool Description

One of the main issues of new drone application is to get the acceptance from the different stakeholders: national or local authorities, politicians, the public, associations. Indeed, the increase presence of drones in the air can be perceived as a threat:

- For passenger transportation since they could disturb manned aircraft,
- For people or critical infrastructure safety on the ground since they could crash on them,
- For people privacy since drone could gather data about people life while flying,
- For people quality of life since drone are very noisy,
- For wildlife preservation since drone flight could disturb preserved reserve.

Any of these threats could justify a decision to forbid a drone operation either by regulation from national or local authorities or by pressure from people or associations.

To ensure acceptance of drone solution by the different stakeholders, it is important to make an early evaluation of how their ConOps would impact every stakeholder involved (users as well as third parties).

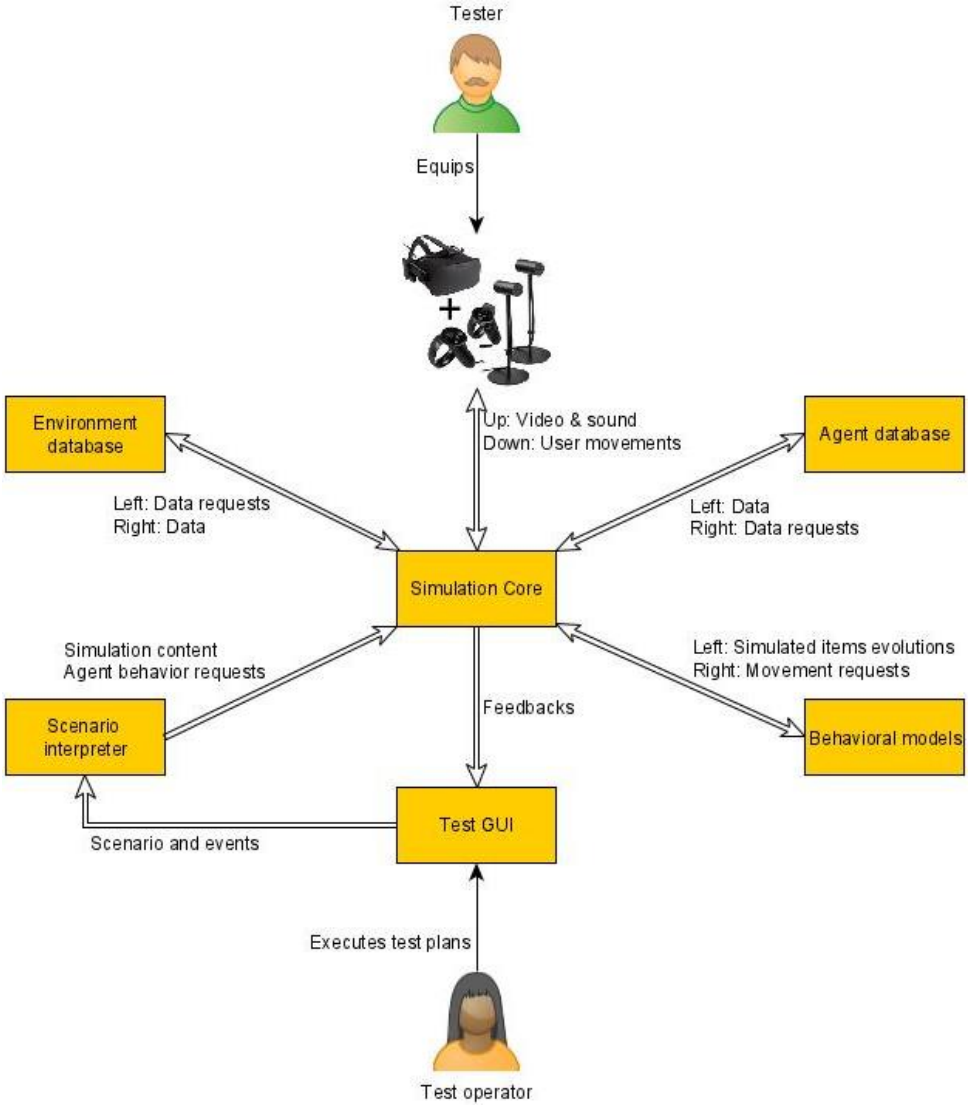


Figure 44 Architecture of the Stakeholder Acceptance Digital Test Bench.

The purpose of the Stakeholder Acceptance Digital Test Bench is to immerse stakeholders into the reality of UAV operations (noise, visual impact, workload for operators...) using AR/VR. That way, its users will be able to evaluate the impact of drone into their airspace/cities by polling the citizen/associations and testing their reactions.

The architecture presented in Figure 44 aims at being able to simulate for a Tester multiple type of operations in various environments with different types of drones. These tests are prepared by a Test Operator who defines scenarios but also injects events during the simulations to provoke different reactions from the Tester.

The modules composing the Acceptance Test Bench are:

- The **Test GUI** is the interface with the test operator that is used to input the scenarios or trigger events as well as to get the feedbacks from the simulation,
- The **Scenario Interpreter** is translating the Test Operator commands into request for the Simulation Core,
- The **Environment Database** stores the different world representations (different city types, industrial infrastructures…) that are used for VR rendering to the Tester,
- The **Agent Database** stores the different agents that will evolve in the simulation. They may be drone but also third parties (other aircraft, ground vehicles…) defined in the scenarios to represent several natures of events (traffic conflicts, interference with other aircraft…),
- The **Behavioral Models** define how each agent evolves in the environment depending on the evolution request defined by the simulation scenarios (waypoints, weather conditions…) and events (failures, encounters…),
- The **Simulation Core** coordinates the simulation, integrates all data and provides the visual and sound rendering to the VR headset used by the Tester.

4.10.2 Improvements planned

The goal of Stakeholder Acceptance Digital Test Bench is to bring the following improvements:

- Allow cities, politicians and associations to evaluate the insertion of drone into the airspace, by enabling survey of citizen opinions, test of reactions…
- Allow drone operators to refine their high-level requirements using this tool.
- Ensure the capture and validation of all the requirements necessary to obtain the acceptance of all the stakeholders.

4.10.3 Functional and non-functional requirements

UC3-DTC-84	The Stakeholder Acceptance Digital Test Bench shall simulate the various environments involved in C4D use cases
UC3-DTC-85	The Stakeholder Acceptance Digital Test Bench shall render the visual and noise signatures of the different vehicles involved in C4D use cases
UC3-DTC-86	The Stakeholder Acceptance Digital Test Bench shall allow the Test Bench users to evolve in the simulated environment
UC3-DTC-87	The Stakeholder Acceptance Digital Test Bench shall allow the definition of use case scenarios by a super user

4.10.4 Specific standard and/or regulation requirements

There are no specific standard and/or regulation requirements.

4.10.5 Specific installation and/or functional requirements

The knowledge of the environments of use in the C4D use cases is supposed to be available as an input to build Environment databases.

AR/VR immersive goggles and headset are required for digital rendering.

4.11 e-Handbook for safety, security and privacy (ALTRAN)

4.11.1 Tool Description

The e-Handbook tool is linked with the D2.5 deliverable that should provide a macroscopic understanding of the regulatory framework and qualification process of drone systems to obtain a Permit to Fly.

The purpose of the e-Handbook is to link operations with applicable regulations and standards regarding safety, security and privacy.

The e-Handbook aims at guiding architects and designers by identifying the minimum set of requirements and Means of Compliance to comply with regulations for their intended use.

The architecture of the e-Handbook depicted in Figure 45 is straightforward. It relies first on an interface that allows a comprehensive description of the ConOps of a given drone use case that is interpreted and linked to databases containing the applicable requirements and corresponding Means of Compliance.

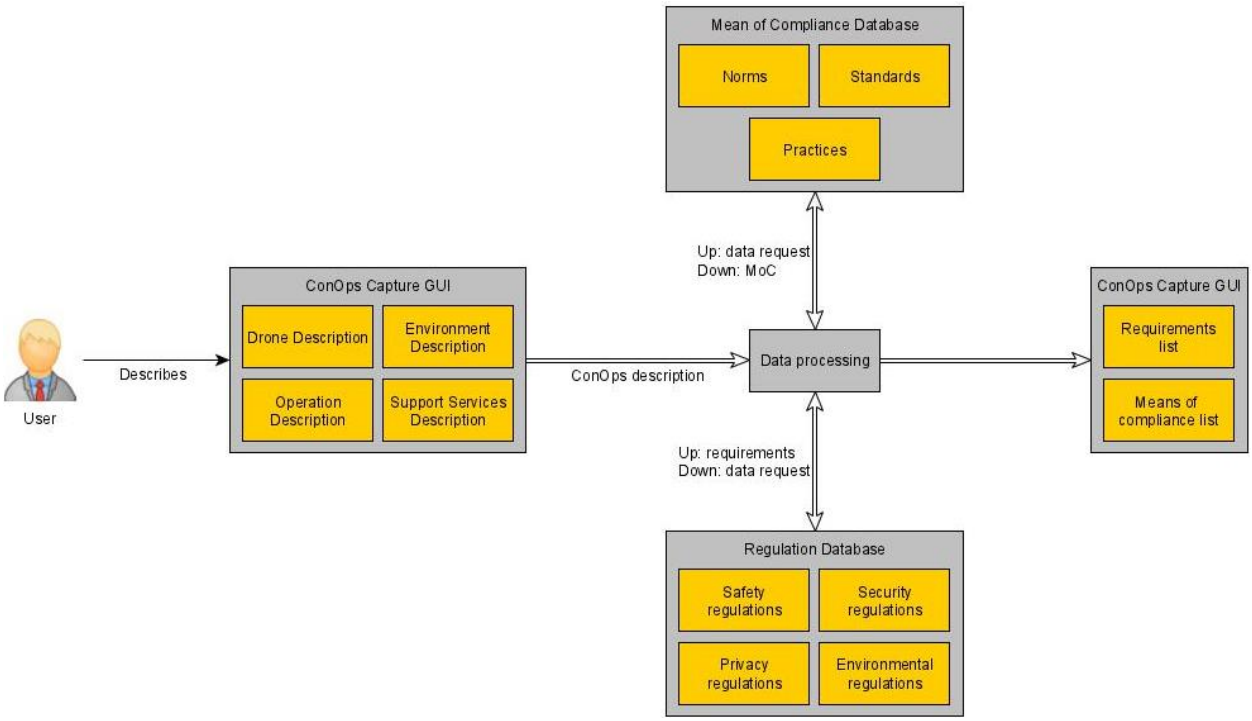


Figure 45 Architecture of the e-Handbook for safety, security and privacy.

4.11.2 Improvements planned

The improvements brought by the e-Handbook are:

- To guarantee that Permit to Fly is obtained,
- To ensure that the development ensures safety, security and privacy at minimum cost,
- To guide the Designers and Architects through an interactive environment (wizard like) towards the results.

4.11.3 Functional and non-functional requirements

UC3-DTC-88	The e-Handbook shall provide applicable regulations and standards regarding safety, security and privacy into a comprehensive database

UC3-DTC-89	The e-Handbook shall link the Concept of Operations items to the regulation database items
UC3-DTC-90	The e-Handbook shall take the definition of Concept of Operations as single needed input from the Designers and Architects

4.11.4 Specific standard and/or regulation requirements

There are no specific standard and/or regulation requirements.

4.11.5 Specific installation and/or functional requirements

The following regulations requirements are mandatory inputs in order to build the Regulation and Mean of Compliance Databases components of the e-Handbook:

- Common delegated regulation EU 2019/945
- Implemented regulation EU 2019/947
- U-space services

4.12 6DOF Test Bench "ARMADA" (ALTRAN)

4.12.1 Tool Description

The 6DOF Test Bench is already developed by ALTRAN (and is known as "ARMADA" test bench).

It consists of two bricks, as shown in Figure 46:

- The first brick is a stand-alone brick and is a test mean on its own. It allows to monitor the performances of multirotor drones and its components. It also provides an answer to higher level needs of developing hybrid energy solutions and optimising aerodynamic and acoustics aspects. This first brick will be called « Monitoring test bench » or « DTBM »
- The second brick is an additional brick and optional. It allows coupling « DTBM » to simulation via a 3D physics and graphics engine.

The set of these two coupled bricks forms a simulation loop « Hardware in the Loop » or « HITL ».

4.12.2 Improvements planned

Planned developments for the HITL simulation platform are the followings:

- Development of a disturbance and failure generator
  - Wind, others, …
  - GPS malfunctions, ESC, engines, …
- Development of scripts emulating a radio control
  - Test reproducibility
  - Compare component performance
  - Compare new control laws
- Integration and testing of innovative functions developed by ARMADA project at ALTRAN
- Adaptation to others Flight Controls (Software)
  - Arducopter
  - Paparazzi
  - Etc…
- Adaptation to others Flight Controllers (Hardware)
  - DJI
  - PARROT
  - Etc…

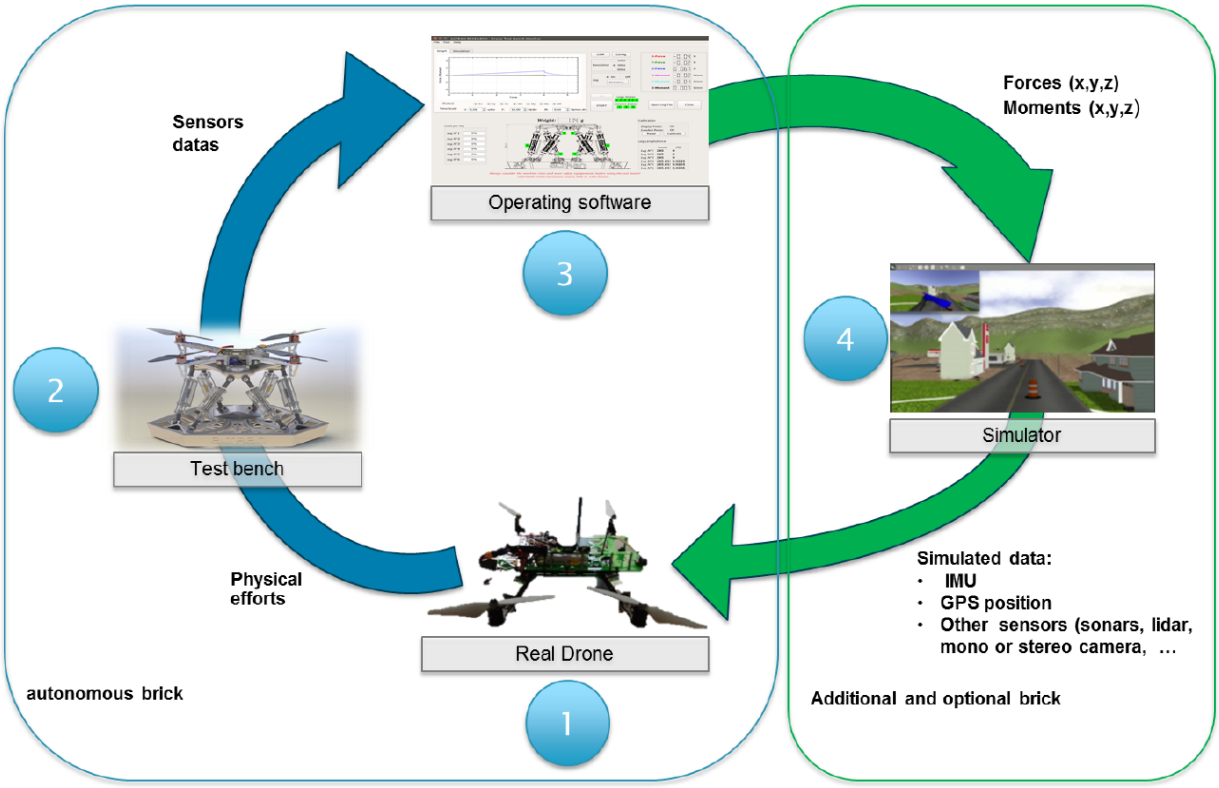


Figure 46 Coupling of the two bricks of ARMADA.

4.12.3 Functional and non-functional requirements

UC3-DTC-91	The 6DOF Test Bench shall sense the forces generated by the multi-rotor drones to be tested in order to create an acceleration
UC3-DTC-92	The 6DOF Test Bench shall allow the user to carry out tests on the take-off and landing phases for a VTOL exclusively
UC3-DTC-93	The 6DOF Test Bench shall accept all types from helicopters to drones with N propellers whether the propulsion system is counter rotating or not.

4.12.4 Specific standard and/or regulation requirements

There are no specific standard and/or regulation requirements.

4.12.5 Specific installation and/or functional requirements

The 6DOF Test Bench is not adapted to the fixed wings drones.

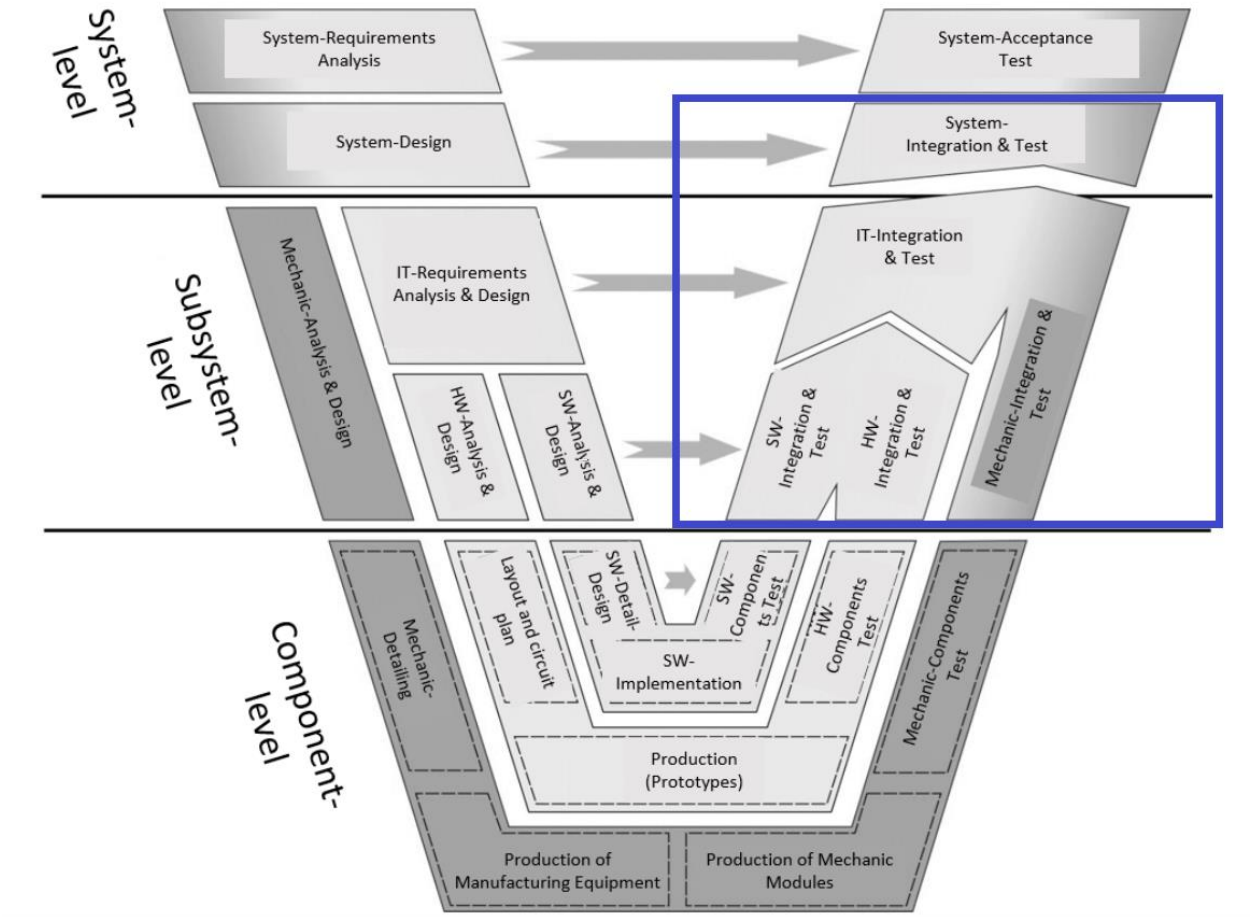


Figure 47 Positioning of the 6DOF Test Bench "ARMADA" in the V-cycle.

5 Conclusions

In this Deliverable, the background technology provided by the partners to C4D is described. All the tools improve the current state-of-the-art in some concrete aspect and its contribution to the global objectives of the project will be assessed in a UC. The requirements imposed in order to qualify and even, quantify these improvements have been collected. Both pieces of information are relevant in order to inform the UCs so that they can take advantage of the reduction in design effort and time they can achieve by integrating these tools.

Each tool has been described in its functionality, the improvements planned, the installation requirements, if any, and the UC where the tool will be assessed regarding the requirements imposed by the users. These improvements will be described in Deliverables D6.2 and D6.3 when the preliminary and final versions of the tools will be provided and the C4D Design Tool Framework (C4D-DTF) made publicly available.

The tool collection covers all the steps in the mechatronic development V-Cycle. The final C4D Design Tool Framework will integrate all them with currently available commercial and open-source tools. This will be done in a flexible way so that the C4D-DTF can be adapted to the different design flows used in each company.

6 ANNEX I

6.1 C4D design technologies.

					4.1 Service Specification (User and Service providers)				4.2 HW/SW System Development Cycle (System and Component Providers)					
ID	**Partner**	**Expected Outcome**	**Expected TRL**	**Lined with UC**	4.1.1 User Requirements	4.1.2 Acceptance Testing	4.1.3 Data Analytics	4.1.4 Mission Planning	4.2.1 System Requirements	4.2.2 Design	4.2.3 Implementation	4.2.4 Integration	4.2.5 Testing (Verification)	4.2.6 Validation
WP6-01	AIT	Extended workflow engine for usage by drone manufacturers	5	UC5						■			■	
WP6-02		AIT Security Analysis for Drones	6	UC5						■				
WP6-03		MoMut Protocol Testing	4	UC5									■	
WP6-04	BUT	Testing Tool set	5	UC4				■						
WP6-05		Big Data analytics	4				■							
WP6-06		Modelling & Simulation Tool	5					■						
WP6-07		Mission design and optimization	4					■						
WP6-08		Safety Analysis Tool	5								■		■	
WP6-09	UWB	ROS/Gazebo modules for autonomous drone battery management simulation and validation	4	UC4					■	■	■	■	■	■

					4.1 Service Specification (User and Service providers)				4.2 HW/SW System Development Cycle (System and Component Providers)					

ID	Partner	Expected Outcome	Expected TRL	Lined with UC	4.1.1 User Requirements	4.1.2 Acceptance Testing	4.1.3 Data Analytics	4.1.4 Mission Planning	4.2.1 System Requirements	4.2.2 Design	4.2.3 Implementation	4.2.4 Integration	4.2.5 Testing (Verification)	4.2.6 Validation
WP6-10	ENAC	Paparazi simulation and real flight autopilot validation tools	5	UC1										
WP6-11	SIEMENS	DRONES components models, Amesim standard interfaces with drones controls and environments	6	UC3										
WP6-12	ENSMA	Model-based framework to component-based drone architectures (including Model-based repository as a designer analysis support and a Schedulability analysis tool)	4	UC3										
WP6-13	UNIMORE	Application development tools for the heterogeneous onboard computing platform	4	UC5										

					4.1 Service Specification (User and Service providers)				4.2 HW/SW System Development Cycle (System and Component Providers)					
ID	**Partner**	**Expected Outcome**	**Expected TRL**	**Lined with UC**	4.1.1 User Requirements	4.1.2 Acceptance Testing	4.1.3 Data Analytics	4.1.4 Mission Planning	4.2.1 System Requirements	4.2.2 Design	4.2.3 Implementation	4.2.4 Integration	4.2.5 Testing (Verification)	4.2.6 Validation
WP6-14	UNISANNIO	Assessment of verification and validation tools for based design and software-in-the-loop and hardware-in-the-loop methodologies	5	UC5				■					■	■
WP6-15	UNISS	Advanced design tool - MDC	4	UC5						■	■	■		
WP6-16		Advanced verification tool – SAGE-VS	4						■				■	
WP6-17	UNIVAQ	Mixed-criticality aware DSE	4	UC5						■	■	■		
WP6-18	ANYWI	Framework and toolkit for validation of robustness of path management for multi-path communication system	5	UC4									■	■
WP6-19		Framework and toolkit for validation of APIs for collection of connection metadata of multi-path communication system	5										■	■

					4.1 Service Specification (User and Service providers)				4.2 HW/SW System Development Cycle (System and Component Providers)					
ID	**Partner**	**Expected Outcome**	**Expected TRL**	**Lined with UC**	4.1.1 User Requirements	4.1.2 Acceptance Testing	4.1.3 Data Analytics	4.1.4 Mission Planning	4.2.1 System Requirements	4.2.2 Design	4.2.3 Implementation	4.2.4 Integration	4.2.5 Testing (Verification)	4.2.6 Validation
WP6-20	ACORDE	ACORDE ESL eSW Design Environment	5	UC2										
WP6-21		Indoor Positioning System Model&Analysis Framework (IPS MAF)	4											
WP6-22	IKERLAN	IoT Environment Extra-functional requirements validation and monitoring toolchain	5	UC2										
WP6-23		Event based IoT Environment validation methodology and toolchain	5											
WP6-24	CEA	Papyrus4Robotics Framework customized for Drones (modelling, code generation and schedulability analysis tools)	5	UC3										

					4.1 Service Specification (User and Service providers)				4.2 HW/SW System Development Cycle (System and Component Providers)					
ID	**Partner**	**Expected Outcome**	**Expected TRL**	**Lined with UC**	4.1.1 User Requirements	4.1.2 Acceptance Testing	4.1.3 Data Analytics	4.1.4 Mission Planning	4.2.1 System Requirements	4.2.2 Design	4.2.3 Implementation	4.2.4 Integration	4.2.5 Testing (Verification)	4.2.6 Validation
WP6-25	UNICAN	S3D: Mixed-Criticality system modelling, design & verification framework	5	UC1 UC3										
WP6-26		SoSSim: Architectural exploration and design tool with extra-functional analysis	6	UC1 UC3										
WP6-27	SM	Design tools for drone mechanical parts and subsystems development for autonomous drone battery management system	4	UC4										
WP6-28	SHERPA	PhiSystem (specific modelling tool for cyber-physical systems) to flight vehicles	3	UC3										

					4.1 Service Specification (User and Service providers)				4.2 HW/SW System Development Cycle (System and Component Providers)					
ID	**Partner**	**Expected Outcome**	**Expected TRL**	**Lined with UC**	4.1.1 User Requirements	4.1.2 Acceptance Testing	4.1.3 Data Analytics	4.1.4 Mission Planning	4.2.1 System Requirements	4.2.2 Design	4.2.3 Implementation	4.2.4 Integration	4.2.5 Testing (Verification)	4.2.6 Validation
WP6-29	ALTRAN	Stakeholder Acceptance Digital Test Bench	1	UC3	■	■								■
WP6-30		e-Handbook for safety, security and privacy	1						■				■	■
WP6-31		6DOF Test Bench "ARMADA"	6									■	■	
WP6-32	ALM	ROS1 & ROS2 infrastructure/ dev-ops.	6	UC4							■	■		
WP6-33		Cloud-based simulation environment	5			■	■	■		■	■	■	■	■

7 ANNEX II

7.1 Design Technology Requirements

REQ ID	Short description
UC4-DTC-01	Tool shall provide high fidelity representation of dynamic models
UC4-DTC-02	Tool shall allow mission planning with specified target
UC4-DTC-03	Tool shall consider realistic weather data with the additional constraints, like available fuel resources.
UC5-DEM10-DTC-04	Accelerators Programming Model SHALL enable offloading capabilities from the host processing system to the HW accelerators mapped on FPGA.
UC5-DEM10-DTC-05	Accelerators Programming Model SHALL enable the configuration of application-specific accelerators mapped on FPGA.
UC5-DEM10-DTC-06	MDC tool will enable the definition of the on board accelerators
UC5-DTC-07	Design Space Exploration should consider mixed-criticality requirements
UC5-DTC-08	System Simulator should integrate SystemC models
UC5-DTC-09	System Simulator should emulate hierarchical (hypervisor-based) scheduling
UC5-DTC-10	Timing Simulator should consider Hypervisors characterization correctly
UC5-DTC-11	Simulation time should not be dependent on the time granularity related to the events generated by the environment (i.e., the test bench)
UC5-DTC-12	Simulation time should be reduced by at least 15% on average case
UC2-DEM1-DTC-13	Basic action, i.e. edition, building, and simulation of models; assisted generation of the implementation software (firmware); and firmware verification integrated in a single graphical environment
UC2-DEM1-DTC-14	Integration of version control system
UC2-DEM1-DTC-15	Models based on standard language/procedures
UC2-DEM1-DTC-16	Support of simulable models, with integrated model of time
UC2-DEM1-DTC-17	Significant improvement of model simulation speed vs Matlab or Octave models (target speed-up=10)
UC2-DEM1-DTC-18	Support of specific libraries for GNSS position&attiude estimation

UC2-DEM1-DTC-19	Support of mechanisms for automating or assisting firmware generation, such the process requires seconds/minutes instead of days/months as it happens on the current manually based generation process
UC2-DEM1-DTC-20	Support for simulation-based verification of the automatically generated firmware, supporting test parallelization.
UC2-DEM1-DTC-21	Simulation based verification faster that running firmware tests on a physical prototype. The objective is speed-up >1 without test parallelization, and scalable verification speed-up with the parallelization of independent tests.
UC2-DEM1-DTC-22	Support of graphical analysis of outputs, e.g. Matlab, Octave, Gtkwave.
UC2-DEM1-DTC-23	Affordable cost of the design environment (<2K€/year)
UC3-DEM02-DTC-24	Reduce the effort required for developing a system component.
UC3-DEM02-DTC-25	Ease the integration and configuration of the system different components.
UC3-DTC-26	S3D will be able to model a complete drone mission.
UC3-DTC-27	From the S3D model a simulation model for the drone mission will be generated.
UC3-DTC-28	The verification cycle using the S3D Model in the Loop (MiL) technology will reduce the model-simulation cycle in more than 10%.
UC4-DTC-29	The tool shall lead the developer through the design process of DP integration
UC4-DTC-30	The design library should unify and simplify the design of DP battery system hardware
DEM9-DTC-31	Development workflow
DEM9-DTC-32	Model-based testing for drone communication
UC4-DTC-33	The simulator shall simulate weather and wind effects.
UC4-DTC-34	The simulator shall allow capturing simulated HDR for HDR multi-exposure assembly.
UC4-DTC-35	The simulator shall emulate refuelling process
UC4-DTC-36	The simulator shall communicate using MAVLink messaging protocol
UC3-DTC-37	Reduce the effort required for integrating a component to the system, either tightly or loosely coupled
UC3-DTC-38	Enable the simulation of several heterogeneous drones with realistic flight models and wind conditions

UC3-DTC-39	Provide control and navigation algorithms, with complex flight plan routines if necessary
UC3-DTC-40	Perform automatic compilation and verification of the code base to provide a safer tool
UC5-DTC-41	The simulator shall simulate the path planner for performance evaluation.
UC5-DEM10-DTC-42	SAGE-VS should support the software criticality category defined for UAV standard (e.g., DO178C)
UC5-DEM10-DTC-43	SAGE-VS can be used to manage critical situations (task interferences, scheduling issues, fault of failure conditions).
UC5-DEM10-DTC-44	SAGE-VS can be used to verify the solutions for the management of critical situations, with improved situation awareness
UC5-DEM10-DTC-45	NeVer will support the design and the verification of AI algorithms used to detect and identify parasite animals and to classify leaf diseases using imaging sensor data.
UC2-DTC-46	The system shall be monitored in order to detect unexpected events
UC2-DTC-47	AsyncAPI toolbox shall allow to define the specification of the asynchronous communication protocol API (obtaining a single point of trust).
UC2-DTC-48	AsyncAPI toolbox shall allow to generate documentation related to the specification of the asynchronous communication protocol API.
UC2-DTC-49	AsyncAPI toolbox shall allow to generate source code related to the specification of the asynchronous communication protocol API.
UC2-DTC-50	AsyncAPI toolbox shall allow to generate tests to detect communication consistency failures.
UC3-DTC-51	The platform shall provide a possibility to simulate drones. To make the simulation environment close to reality, the parameters related to drones (aerodynamics, propulsion, wind etc.) shall be modifiable.
UC3-DTC-52	The platform shall simulate the sensors (Lidars, Camera) and the surrounding environment (buildings, trees etc. ...)
UC3-DTC-53	The platform shall allow to test different controllers and navigation algorithms with minimal efforts
UC4-DTC-54	The toolkit shall test and validate that available communication links via different operators (in case of mobile connections) or access technologies are discovered and made available to the path management system of the multipath communication system on the drone.

UC4-DTC-55	The toolkit shall validate the collection of metadata from data connections between drone and ground.
UC5-DEM9-DTC-56	Safety-security co-analysis
UC5-DEM9-DTC-57	The communication between land-bound sensors and drones shall be trusted
UC5-DEM9-DTC-58	The communication between drones and base station shall be trusted
UC4-DTC-59	Verification tool for drone software which perform static analysis of pieces of software that are safety or otherwise critical
UC4-DTC-60	The tool shall provide data analysis capabilities. It shall be possible to detect different objects from video/images.
UC4-DTC-61	The Big Data Analysis tool should unify and simplify the processing of acquired data.
UC3-DTC-62	The tool shall provide system simulation integration capabilities. It shall be possible to integrate different physical domains as well as integrate third party tools.
UC3-DTC-63	System modeling time shall be reduced by 20% with respect to current methods
UC3-DTC-64	The tool shall provide a modular approach to system modeling. The goal is to allow models reusability, minimizing the time needed to adapt existing models to simulate novel drone configurations. The adapting time shall be reduced by 20% with respect to current methods.
UC3-DTC-65	MoSaRT shall reduce the design and analysis efforts
UC3-DTC-66	MoSaRT shall allow designers to check the schedulability of their models without being expert
UC3-DTC-67	MoSaRT shall allow to compare different analysis results providing by different analysis tools
UC2-DEM2-DTC-68	Integration on graphical environment for edition, building, and simulation of indoor positioning system models
UC2-DEM2-DTC-69	Integration of version control system on the IPS MAF
UC2-DEM2-DTC-70	Support modelling of based on standard language/procedures
UC2-DEM2-DTC-71	Support of simulable models, with integrated model of time
UC2-DEM2-DTC-72	Support high-level modelling (effects on latencies) of medium access protocols
UC2-DEM2-DTC-73	Interface for describing the deployment of the UWB anchors

UC2-DEM2-DTC-74	IPS model output which can be directly passed to 3rd party tools for user friendly graphical analysis, e.g. Matlab, Octave, Gtkwave.
UC2-DEM2-DTC-75	No license costs
UC2-DEM2-DTC-76	Scalable simulation time, allow fluent design of small models (e.g. minutes max. for less than 10 anchor models), few hours for a complex model (dozens anchors, with the highest level of detail on the medium access level, the complete flight simulation, including warm-up and normal run phases).
UC2-DTC-77	SoSim will be able to simulate a complete drone mission. The simulation model will be automatically generated from the S3D model
UC2-DTC-78	SoSim will support performance estimation of non-functional metrics such as energy and delays
UC2-DTC-79	SoSim will accelerate performance analysis in more than 20%.
UC4-DEM2-DTC-80	(UC4-PRF-12) COTS standards for communication: ROS2, MAVlink, WIFI, etc.
UC4-DEM2-DTC-81	(UC4-PRF-05) Common model creation: SLAM (SLAM Modelling and Simulation) (semantical rich modelling)
UC4-DEM2-DTC-82	(UC4-PRF-10) User Interfaces: High level mission control
UC4-DEM2-DTC-83	(UC4-PRF-11) User Interfaces: Live model visualization
UC3-DTC-84	The Stakeholder Acceptance Digital Test Bench shall simulate the various environments involved in C4D use cases
UC3-DTC-85	The Stakeholder Acceptance Digital Test Bench shall render the visual and noise signatures of the different vehicles involved in C4D use cases
UC3-DTC-86	The Stakeholder Acceptance Digital Test Bench shall allow the Test Bench users to evolve in the simulated environment
UC3-DTC-87	The Stakeholder Acceptance Digital Test Bench shall allow the definition of use case scenarios by a super user
UC3-DTC-88	The e-Handbook shall provide applicable regulations and standards regarding safety, security and privacy into a comprehensive database
UC3-DTC-89	The e-Handbook shall link the Concept of Operations items to the regulation database items
UC3-DTC-90	The e-Handbook shall take the definition of Concept of Operations as single needed input from the Designers and Architects
UC3-DTC-91	The 6DOF Test Bench shall sense the forces generated by the multi-rotor drones to be tested in order to create an acceleration

UC3-DTC-92	The 6DOF Test Bench shall allow the user to carry out tests on the take-off and landing phases for a VTOL exclusively
UC3-DTC-93	The 6DOF Test Bench shall accept all types from helicopters to drones with N propellers whether the propulsion system is counter rotating or not.