

 **CATIA** | *No Magic*

2019  **CYBER EXPERIENCE  
SYMPOSIUM**  
MBSE

# system specification development during modeling

Raphael Faudou, Samares Engineering

Hamburg, 2019



# A few words on Samares Engineering

- Research and Consultancy on Systems Engineering

- Requirement engineering
- Model-Based Systems Engineering
- Co-simulation



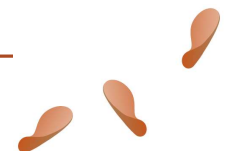
- Involved in different domains



- Strong links with high schools and research institutes

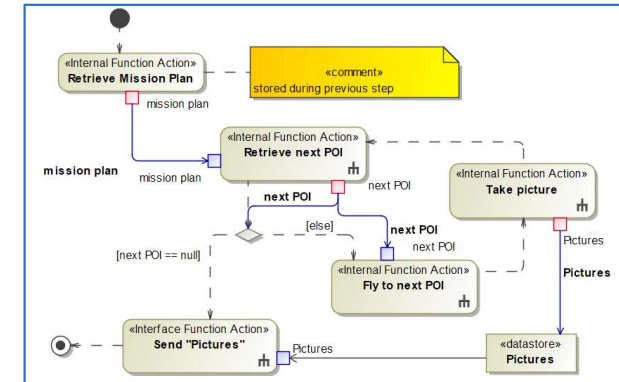
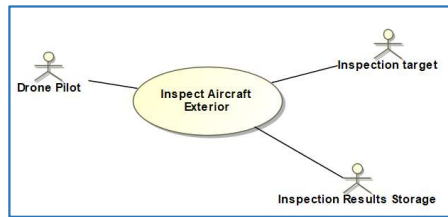


- Context and purpose of the presentation
  - One SysML model can support a lot of Systems Engineering activities...
  - Many views to reconcile...
  - When done, many requirements can be deduced from modeling activities...
- Demos
  - A. Identification of top-level system requirements during needs analysis
  - B. Identification of lower-level requirements during internal architecture
- Way forward
- Q&A



# Modeling can support a lot of SE activities

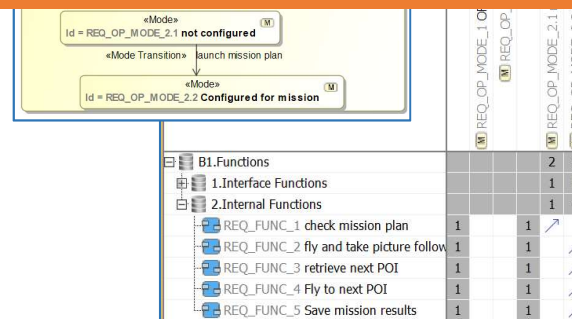
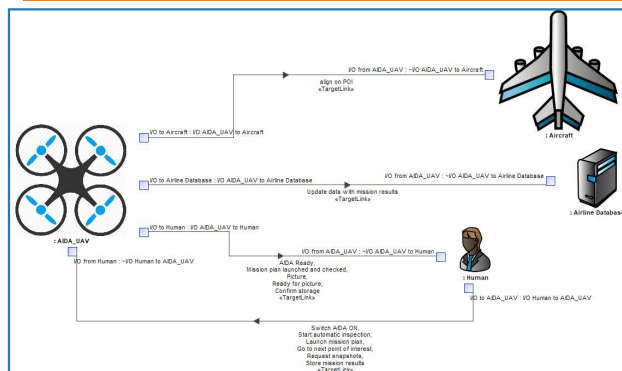
#	△ Name	
1	6 Aircraft inspection feature	There is need for aircraft inspection by UAV
2	7 Automatic or manual	Need for both automatic and manual inspection
3	8 check of mission plan	Mission plan shall be checked before use



9	3	Wind speed limit	UAV shall not fly when wind speed is greater than 30 Km/h
10	3	Wind speed limit	UAV shall not fly when wind speed is greater than 30 Km/h
11	3	Wind speed limit	UAV shall not fly when wind speed is greater than 30 Km/h
12	5	Temperature range	UAV shall be able to perform inspection for temperatures between 40°C
13	4	max Weight	UAV weight shall not exceed 2 Kgs

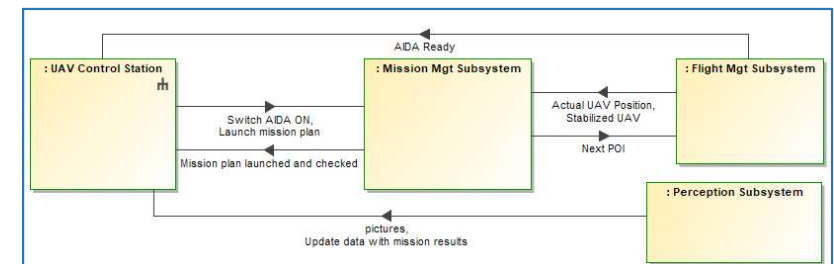
Legend	Allocate	1.1.Logical Con	Flight Mgt Sub	Mission Mgt Sub	Perception Sub	UAV Control Sub
REQ_IF_FUNC_16 Send "Pictures"		1				
REQ_IF_FUNC_25 Receive pictures		1				
2.Internal Functions						
REQ_FUNC_1 Check mission plan		1				
REQ_FUNC_2 Retrieve next POI		1				

Different kinds of diagrams are useful to capture the different (external and internal) views of the system... but how to ensure we build and keep a consistent system definition over time?



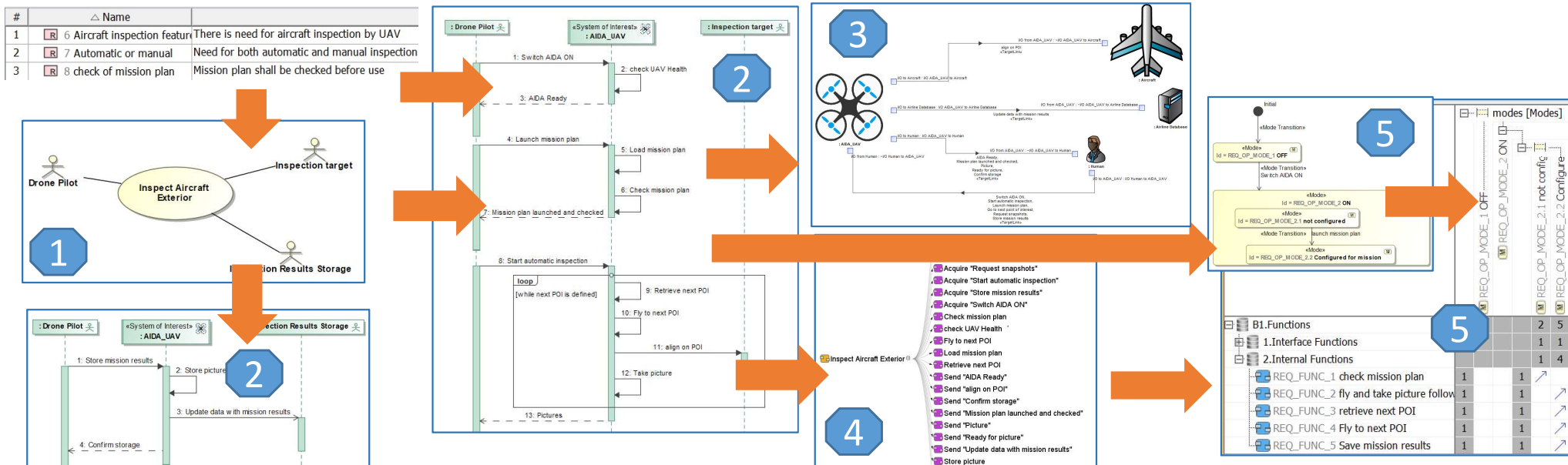
B1 Functions	REQ_OP_MODE_1.0	REQ_OP_MODE_2.1	REQ_OP_MODE_2.2
1.Interface Functions			2 5
2.Internal Functions			1 1
REQ_FUNC_1 check mission plan	1		1
REQ_FUNC_2 fly and take picture follow	1	1	
REQ_FUNC_3 retrieve next POI	1		1
REQ_FUNC_4 Fly to next POI	1		1
REQ_FUNC_5 Save mission results	1		1

REQ_FUNC_10 Return to Ground Station						
REQ_FUNC_11 Stabilize UAV			1			
REQ_FUNC_12 Supervise Mission						
REQ_FUNC_22 Retrieve mission plan			1			



# A. Analysis and structure of needs (1)

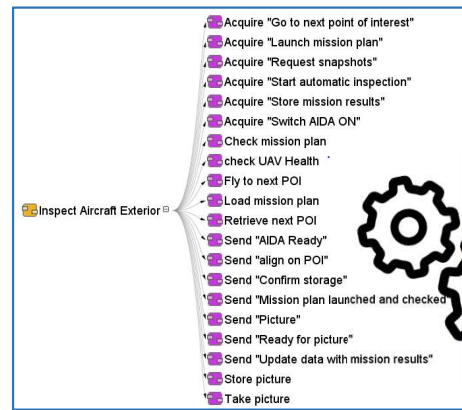
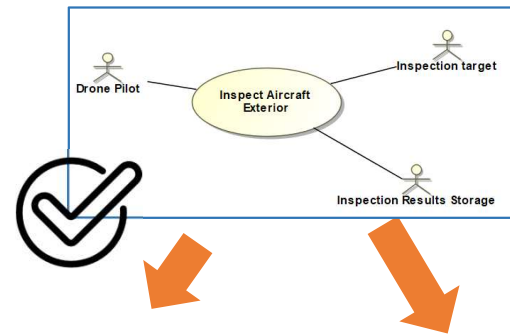
- Many views and concepts to reconcile...
  1. Use Cases define system missions captured from stakeholder needs
  2. Operational scenarios show system interactions and expected reactions
  3. System context synthetizes all external functional interfaces
  4. Top level functions come from operational scenarios or upper level
  5. Operational modes provide boundaries for activation of functions



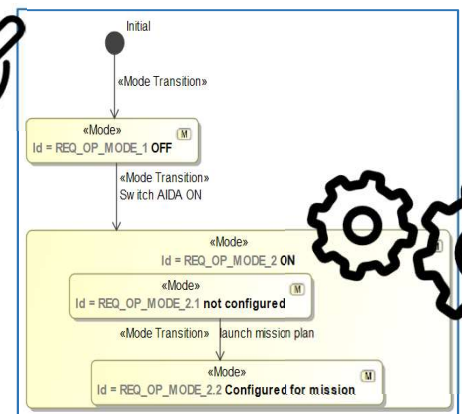
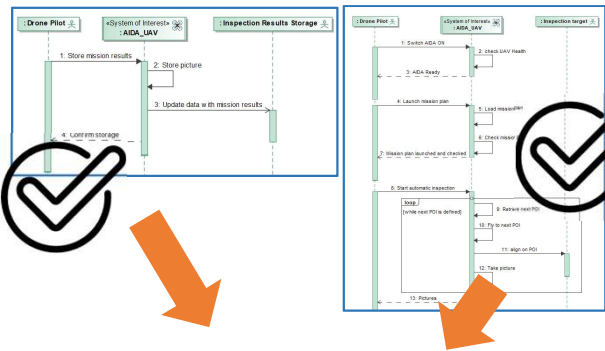
We can define semantic mappings between SysML concepts to ensure consistency

# A. Analysis and structure of needs (2)

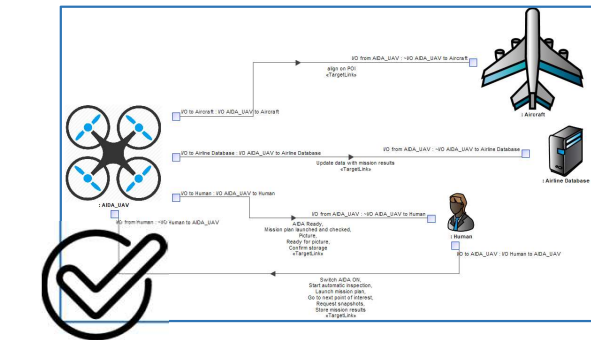
- From all those views, system requirements can be developed



Some requirements are model-based requirements, other remain textual requirements



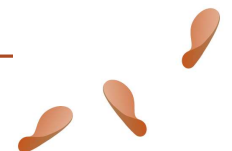
#	Name	Text
28	REQ_IF_FUNC_14 Send "Confirm storage"	The AIDA_UAV shall Send "Confirm storage" to the Drone Pilot.
29	REQ_IF_FUNC_14.1 Confirm storage	The function Send "Confirm storage" shall consume Confirm storage.
30	2.Internal Functions	
31	REQ_FUNC_1 Check mission plan	The AIDA_UAV shall Check mission plan.
32	REQ_FUNC_2 Retrieve next POI	The AIDA_UAV shall Retrieve next POI.
33	REQ_FUNC_3 Fly to next POI	The AIDA_UAV shall Fly to next POI.
34	REQ_FUNC_4 Load mission plan	The AIDA_UAV shall Load mission plan.
35	REQ_FUNC_5 check UAV Health	The AIDA_UAV shall check UAV Health.
36	REQ_FUNC_6 Take picture	The AIDA_UAV shall Take picture.
37	REQ_FUNC_7 Store picture	The AIDA_UAV shall Store picture.
38	External Ports [2.System Int...	
43	Physical Characteristics Requ...	
44	5 Temperature range	UAV shall be able to perform inspection for temperatures between -10°C and 40°C
45	Policies and Regulation Requ...	
46	3 Wind speed limit	UAV shall not fly when wind speed is greater than 30 Km/h
47	4 Max Weight	UAV weight shall not exceed 2 Kgs.
48	REQ_MS_1 Modes	The AIDA_UAV shall be able to operate in the following exclusive Modes:
		<ul style="list-style-type: none"> <li>OFF</li> <li>ON</li> </ul>
49	modes	
50	REQ_OP_MODE_2 ON	When the AIDA_UAV is in ON mode, then the AIDA_UAV shall be able to operate in the following exclusive sub-modes:
51		<ul style="list-style-type: none"> <li>Not configured</li> <li>Configured for mission</li> </ul>
52	REQ_TR_MODE_4	When the AIDA_UAV is in Not configured mode, and the signal Launch mission plan is received then the AIDA_UAV shall transition to Configured for mission mode.



When all semantic mappings have been defined, all views contribute to the validation of needs or elaboration of functional requirements. Any change in Sequence Diagram is immediately reflected in other views including requirements (agile approach)

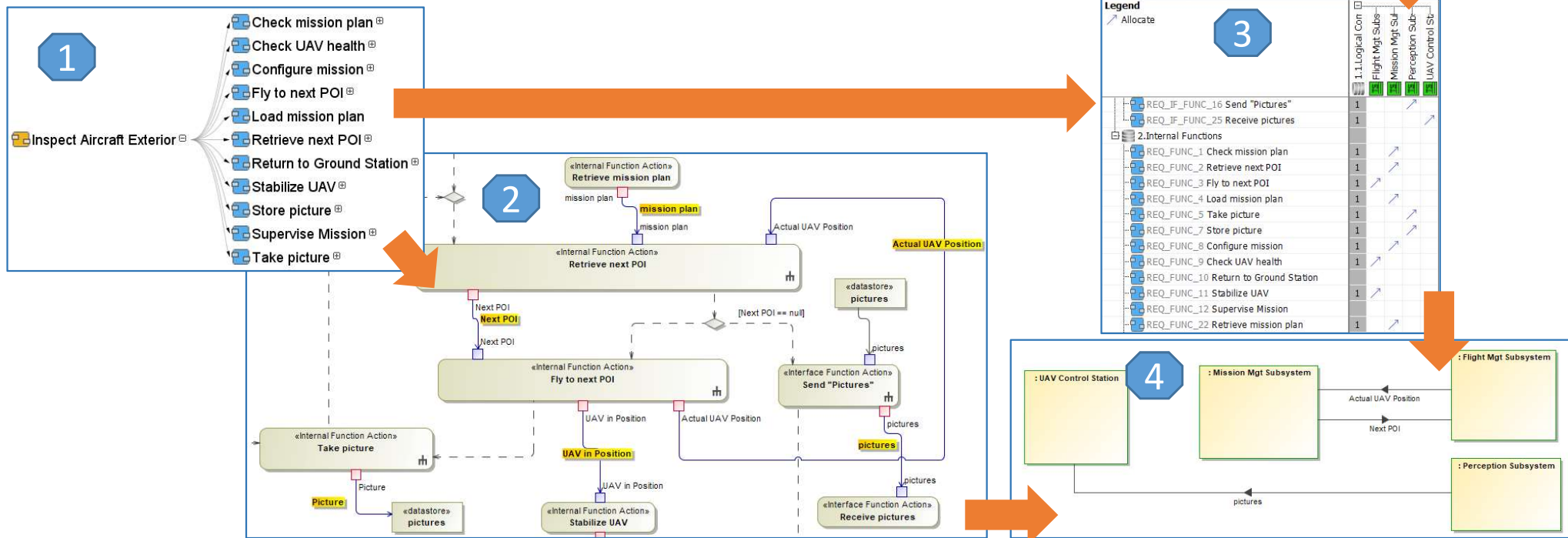
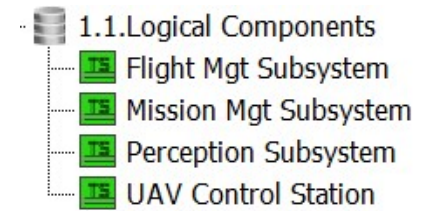
# DEMO 1 – Identification of top-level system requirements during analysis of needs

1. Creation of operational scenarios
2. Synthesis of all external functional interfaces
3. Identification of first modes and transitions
4. **Identification of top-level system requirements**



# B. Functional and logical architecture (1)

- Still many views and concepts to reconcile...
  1. Decompose internal functions
  2. Arrange sub-functions (functional architecture)
  3. Allocate sub-functions on sub-systems/components
  4. Arrange sub-systems/components (logical architecture)



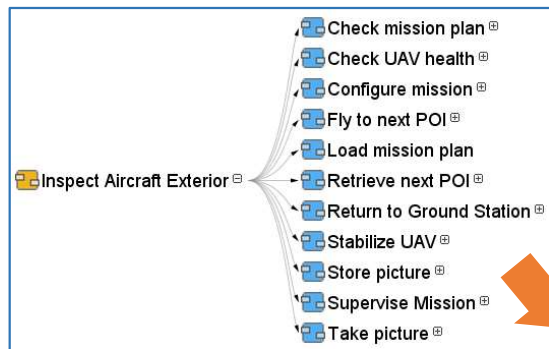
We can define semantic mappings between SysML concepts to ensure consistency



# B. Functional and logical architecture (2)

- From all those views, system requirements can be developed down to subsystem needs (according to logical architecture)


When all semantic mappings are defined, logical architecture can be maintained from functional flows and function allocations : it becomes also possible to initialize subsystem needs document



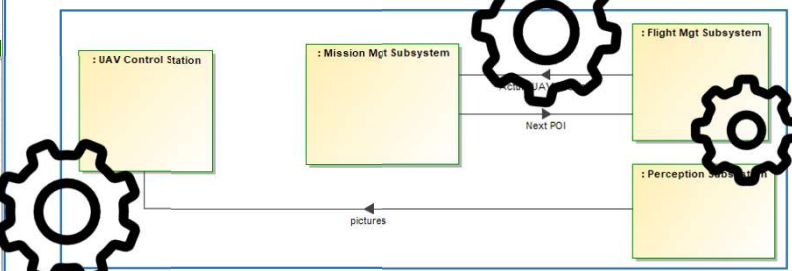
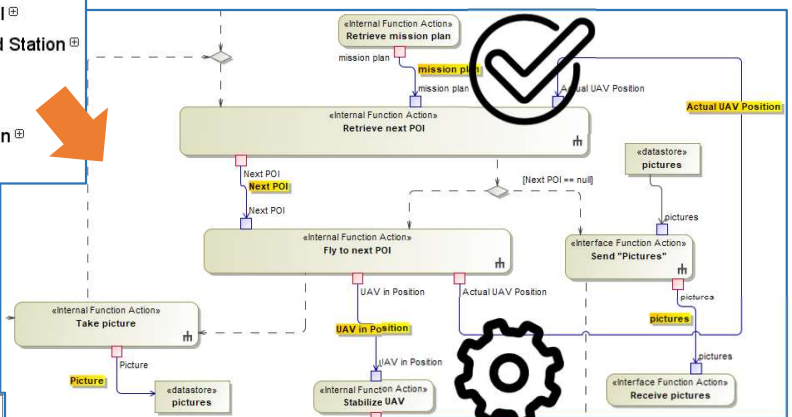
- 1.1. Logical Components
- Flight Mgt Subsystem
  - Mission Mgt Subsystem
  - Perception Subsystem
  - UAV Control Station

Legend

Allocate



Legend	1.1. Logical Con	1.1. Logical Subs	Flight Mgt Subs	Mission Mgt Subs	Perception Subs	UAV Control Sta
REQ_IF_FUNC_16 Send "Pictures"	1					
REQ_IF_FUNC_25 Receive pictures	1					
2. Internal Functions						
REQ_FUNC_1 Check mission plan	1					
REQ_FUNC_2 Retrieve next POI	1					
REQ_FUNC_3 Fly to next POI	1					
REQ_FUNC_4 Load mission plan	1					
REQ_FUNC_5 Take picture	1					
REQ_FUNC_7 Store picture	1					
REQ_FUNC_8 Configure mission	1					
REQ_FUNC_9 Check UAV health	1					
REQ_FUNC_10 Return to Ground Station	1					
REQ_FUNC_11 Stabilize UAV	1					
REQ_FUNC_12 Supervise Mission	1					
REQ_FUNC_22 Retrieve mission plan	1					



Expression of needs for Flight Mgt Subsystem V1.0

1) Context of Flight Mgt Subsystem

Figure 1: Logical Architecture

2) Functions allocated to Flight Mgt Subsystem

The following functions have been allocated to Flight Mgt Subsystem:

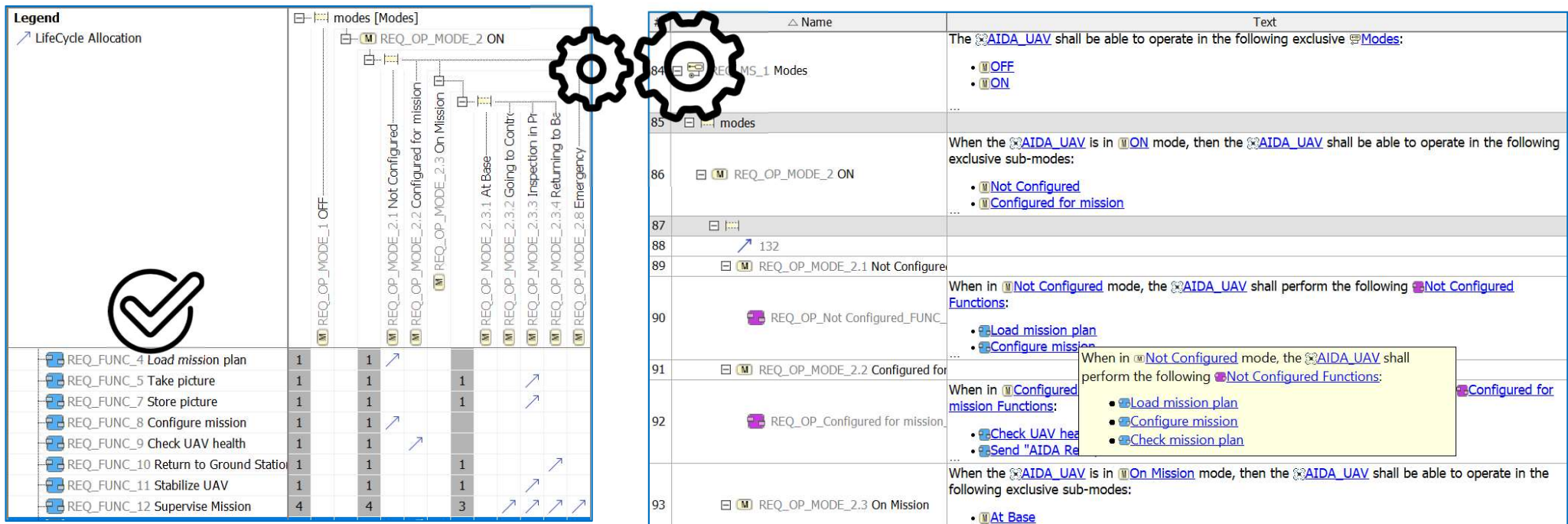
- Fly to next POI
- Check UAV health
- Return to Ground Station
- Stabilize UAV

3) Requirements to be satisfied by Flight Mgt Subsystem

ID	Text	Maturity	Author	Last Date	Modification
REQ_FUNC_10	The AIDA_UAV shall Return to Ground Station.	Medium			
REQ_FUNC_11	The AIDA_UAV shall Stabilize UAV	Medium			
REQ_FUNC_11.1	The function 'Stabilize UAV' shall consume 'UAV in Position'.				
REQ_FUNC_11.2	The function 'Stabilize UAV' shall produce 'Stabilized UAV'.				
REQ_FUNC_3	The AIDA_UAV shall Fly to next POI.	Medium		Wed Oct 09 16:07:41 CEST 2019	
REQ_FUNC_3.1	The function 'Fly to next POI' shall consume 'Next POI'.				

# B. Functional and logical architecture (3)

- Thanks to the contribution of all views, system requirements improve progressively in maturity



**Legend**

LifeCycle Allocation

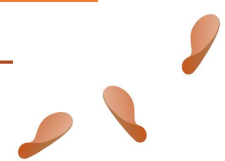
Req ID	Req Name	Req Type	Req Status	Req Priority	Req Category	Req Sub-category	Req Sub-sub-category	Req Sub-sub-sub-category
REQ_FUNC_4	Load mission plan	1	1	1				
REQ_FUNC_5	Take picture	1	1	1				
REQ_FUNC_7	Store picture	1	1	1				
REQ_FUNC_8	Configure mission	1	1	1				
REQ_FUNC_9	Check UAV health	1	1	1				
REQ_FUNC_10	Return to Ground Station	1	1	1				
REQ_FUNC_11	Stabilize UAV	1	1	1				
REQ_FUNC_12	Supervise Mission	4	4	3				

**modes [Modes]**

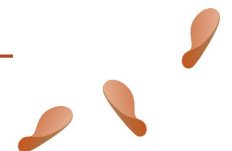
- REQ\_OP\_MODE\_1 OFF
- REQ\_OP\_MODE\_2 ON
  - REQ\_OP\_MODE\_2.1 Not Configured
  - REQ\_OP\_MODE\_2.2 Configured for mission
  - REQ\_OP\_MODE\_2.3 On Mission
    - REQ\_OP\_MODE\_2.3.1 At Base
    - REQ\_OP\_MODE\_2.3.2 Going to Contr
    - REQ\_OP\_MODE\_2.3.3 Inspection in Pr
    - REQ\_OP\_MODE\_2.3.4 Returning to Be
    - REQ\_OP\_MODE\_2.3.8 Emergency

#	Name	Text
84	REQ_MS_1 Modes	The AIDA_UAV shall be able to operate in the following exclusive Modes: <ul style="list-style-type: none"><li>OFF</li><li>ON</li></ul>
85	modes	When the AIDA_UAV is in ON mode, then the AIDA_UAV shall be able to operate in the following exclusive sub-modes:
86	REQ_OP_MODE_2 ON	<ul style="list-style-type: none"><li>Not Configured</li><li>Configured for mission</li></ul>
87		
88	132	
89	REQ_OP_MODE_2.1 Not Configure	When in Not Configured mode, the AIDA_UAV shall perform the following Not Configured Functions: <ul style="list-style-type: none"><li>Load mission plan</li><li>Configure mission</li></ul>
90	REQ_OP_Not Configured_FUNC	
91	REQ_OP_MODE_2.2 Configured for	When in Not Configured mode, the AIDA_UAV shall perform the following Not Configured Functions: <ul style="list-style-type: none"><li>Load mission plan</li><li>Configure mission</li></ul>
92	REQ_OP_Configured for mission	When in Configured mission Functions: <ul style="list-style-type: none"><li>Check UAV hea</li><li>Send "AIDA Re</li><li>Check mission plan</li></ul>
93	REQ_OP_MODE_2.3 On Mission	When the AIDA_UAV is in On Mission mode, then the AIDA_UAV shall be able to operate in the following exclusive sub-modes: <ul style="list-style-type: none"><li>At Base</li></ul>

When allocated to modes, functions are constrained in their usage: there is a new system requirement that defines their validity with regards to system modes...



- Initial conditions: we start from functional architecture
  1. Allocation of sub-functions on sub-systems/components and derivation of logical architecture
  2. Allocation of functions to modes
  3. Initialize expression of needs for sub-systems / component (Document generated)



- The road is still long...
  - Improve modes identification,
  - Complete logical architecture with control flow,
  - Address other constraints (physical, safety...)
  - ...

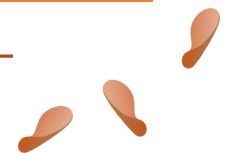


- ... but we know we go in the right direction
- Good feedback on current usage of plugins



- Now we need more UC to progress faster
  - Many domains and many contexts...

Decision to launch a “Live specification partnership program”

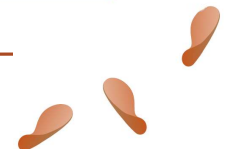


- General principles of this partnership
  - Agreement signed for a given period (6 months / 1 year / 2 years)
  - Samares Engineering provides “Live mode” plugins under NDA
  - Samares’ partner provides industrial Use Cases under NDA
  - Both partners share feedback in use of plugins on industrial UC
  - Samares’ partner benefits from new revisions of plugins
  - Possibility to acquire plugins or order specific improvements
- Launch planned on **18<sup>th</sup> November 2019**
  - Number of partners will be limited (5 to 7, still to be refined)



Want to know more? Express your interest at

[LiveSpecificationPartnership@samares-engineering.com](mailto:LiveSpecificationPartnership@samares-engineering.com)



Thank you for your attention  
Any question before the bell ?

