IRT Saint Exupéry LIV- S085L02-025
IRT SystemX ISX-S2C-DOC-438
Version: V6

Behavioral Cross Check method

DATE: 28/10/2022

Summary

This document aims to explicit and to explain the designed method called "Behavior Cross Check" (BCC). This method is introduced by the top document framed by the S2C project.

This method is based on the concept of "Cross Check of observations" done on the simulation results of two distinct models. The document includes the description of the proposed method, two proof of concept (PoC) based upon SE and SA models to experiment and validate the method and the conclusions of the method.

Author(s)	*Function(s) & name(s)*	*Research engineers* *IRT Saint Exupéry*	*S. Guilmeau*
Checker(s)	*Function(s) & name(s)*	*Head of Systems Engineering* *Centre of Competence* *IRT Saint Exupéry*	*J. Baclet*
Approver	*Function & name*	*Project leader* *IRT Saint Exupéry*	*J. Perrin*

Table of Contents

Evolutions

Table of figures

Table of tables

1 Introduction

1.1 Purpose of document

6

1.2 Referenced documents

1.2.1 S2C reference documents

Title	Reference

1.2.2 External reference documents

Title	Reference

2 Objectives, constraints and Context

2.1 Objectives

-
-
-

2.2 Constraints

2.2.1 Common constraints

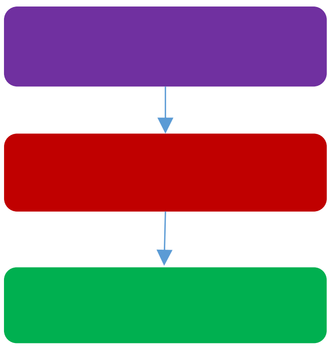
2.3 Context

2.3.1 Common Context

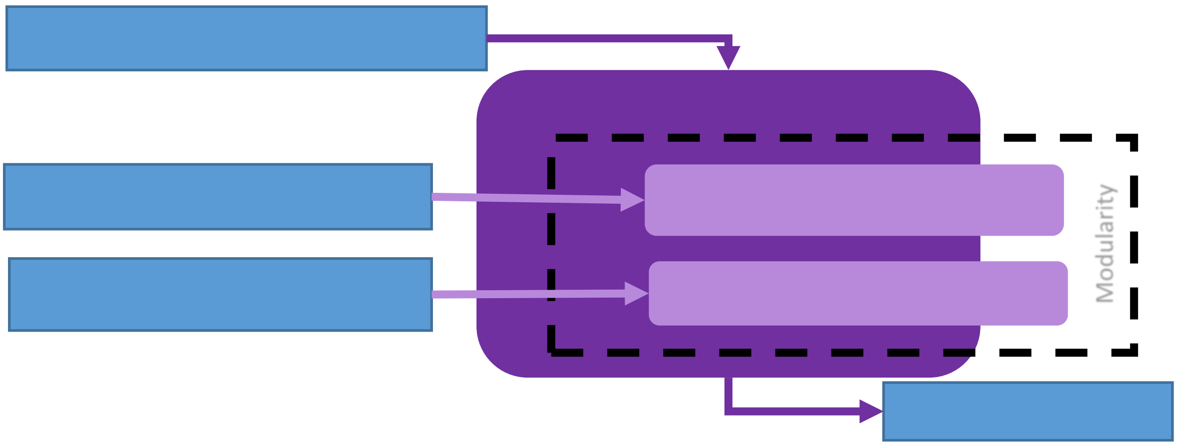
2.3.2 Specific Context

3 Developed method

3.1 BCC Process



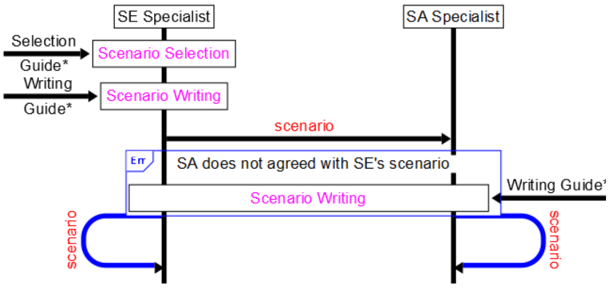
3.1.1 Scenario initialization

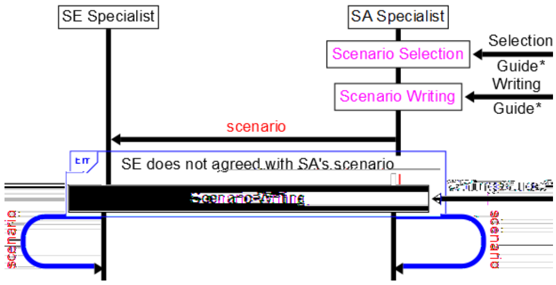


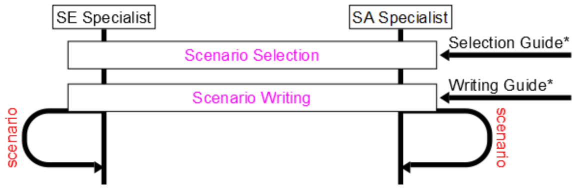
(a) Scenario selection

				#id-d	#id-e	#id-f	#id-g			#id-j	#id-k	#id-l	#id-m	#id-n		
	✓						✓	✓	✓	✓	✓	✓	✓	✓	✓	

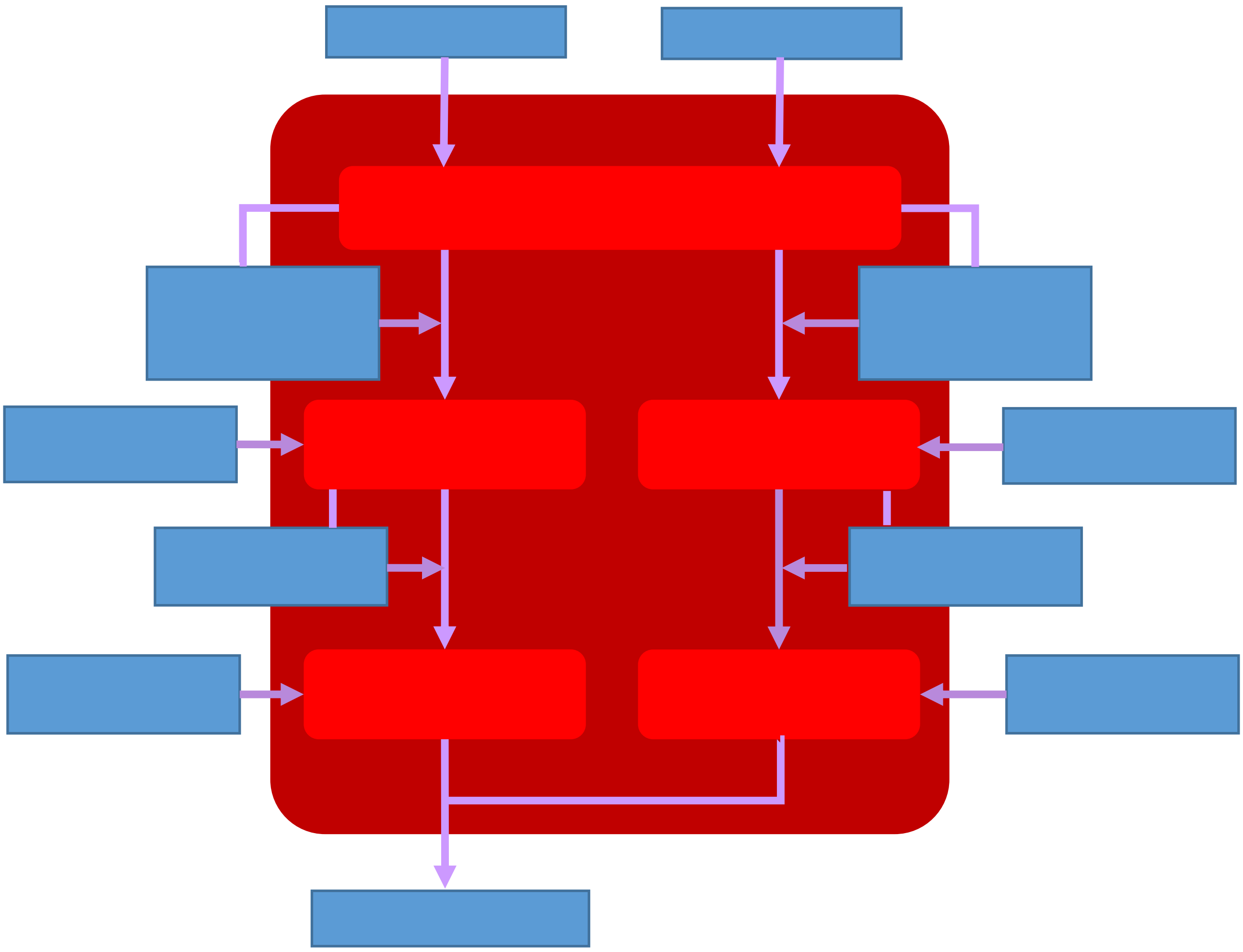
(c) Modularity







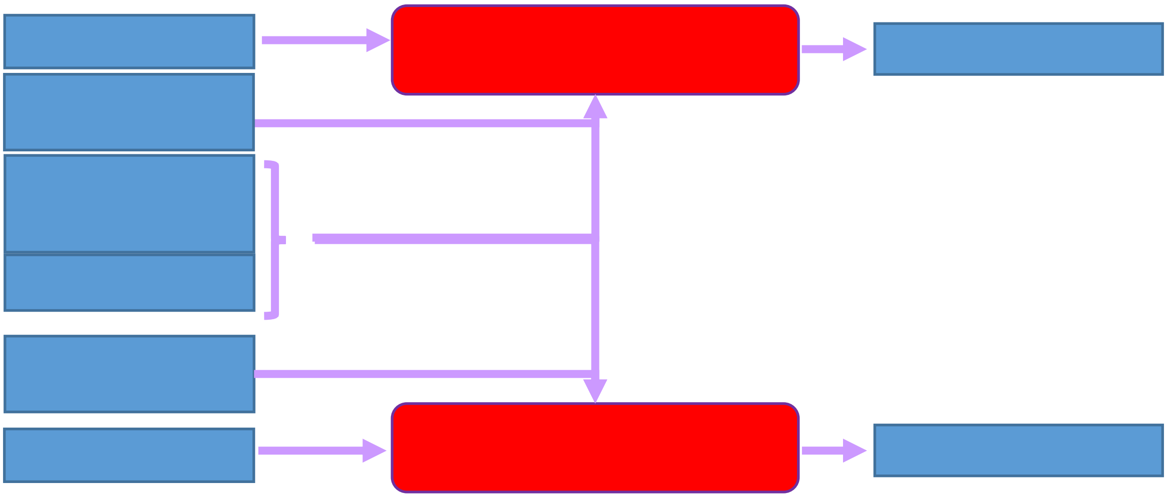
3.1.2 Scenario exploitation



(a) Upstream coordination between Scenario Concepts vs. Domain Variables

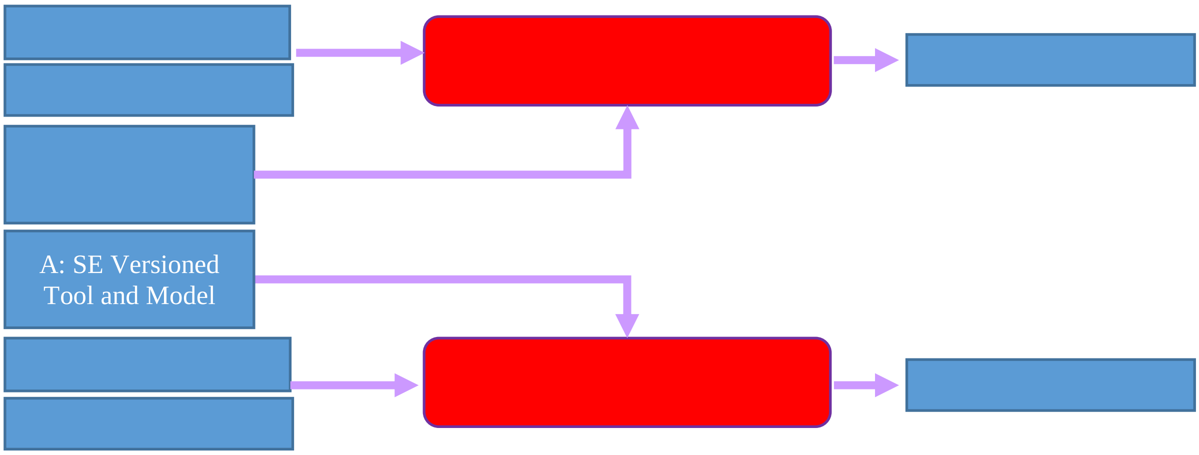
	✓		✓	✓	✓		✓	✓		✓	✓	✓		✓	✓	✓

(b) Transformation from Scenario to Procedure



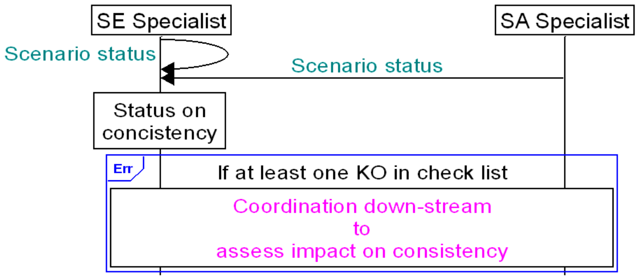
	✓		✓		✓			✓		✓	✓	✓		✓	✓	

(c) Procedure execution



	✓	✓	✓		✓		✓	✓		✓	✓	✓		✓	✓	

3.1.3 Scenario conclusion



	✓		✓	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓	

3.2 Intra-process validation activities

3.3 Iterations

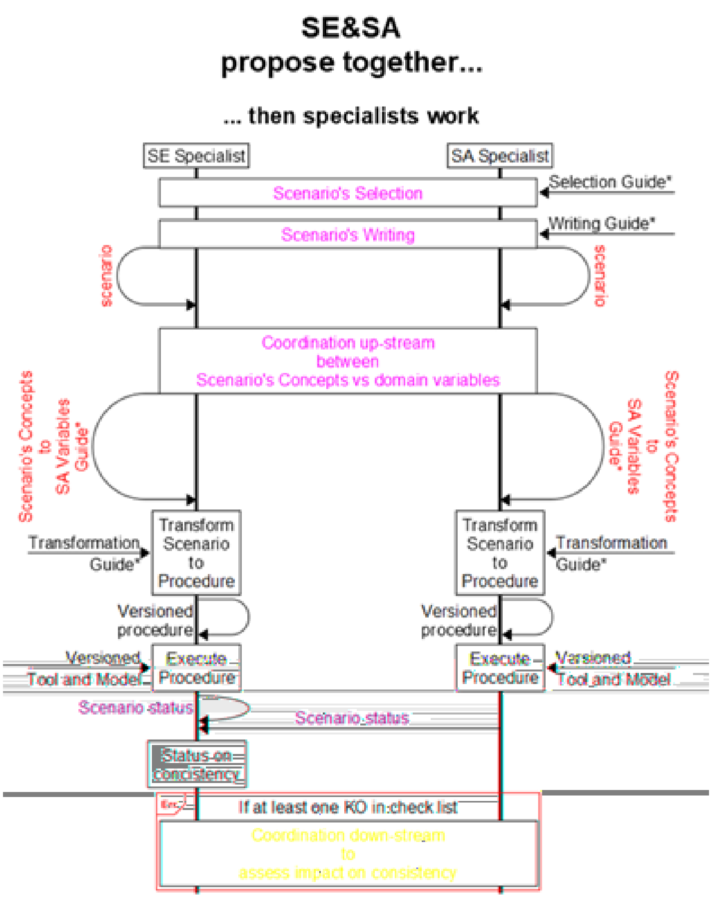
4 Deductible facts before PoCs are done

5 PoC activities

-
-

	[redacted]			[redacted]		
	[redacted]			[redacted]		
	[redacted]			[redacted]		
	[redacted]			[redacted]		
	[redacted]			[redacted]		
	[redacted]			[redacted]		

#sq-0


SE&SA
propose together...
... then specialists work
SE Specialist
SA Specialist
Selection Guide*
Scenario's Selection
Writing Guide*
Scenario's Writing
scenario
scenario
Coordination up-stream
between
Scenario's Concepts vs domain variables
Scenario's Concepts
to
SA Variables
Guide*
Scenario's Concepts
to
SA Variables
Guide*
Transformation
Guide*
Transform
Scenario
to
Procedure
Transform
Scenario
to
Procedure
Transformation
Guide*
Versioned
procedure
Versioned
procedure
Versioned
Tool and Model
Execute
Procedure
Execute
Procedure
Versioned
Tool and Model
Scenario status
Scenario status
Status on
consistency
If at least one KO in check list
Coordination down-stream
to
assess impact on consistency

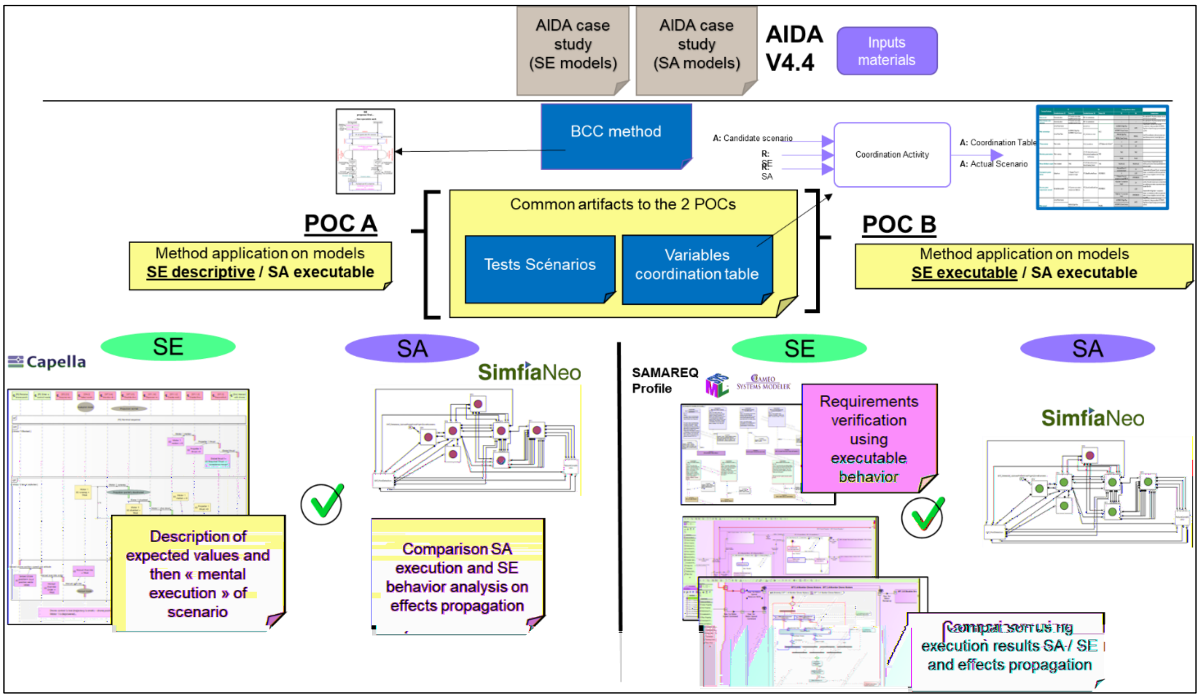
5.1 Proof-of-concept

5.1.1 Preliminary input material

5.1.2 PoCs overview

PoC A

PoC B



5.1.3 Common activity 1– Scenario Initialization

Scenario initialization

(a) Scenario 1: Loss of motor

(b) Scenario 2: Loss of Attitude Information

(c) Scenario 3: Position Information Erroneous

(d) Scenario 4: Switch From Auto to Manual Mode

5.1.4 Common activity 2 - Upstream coordination between Scenario Concepts vs. Domain Variables

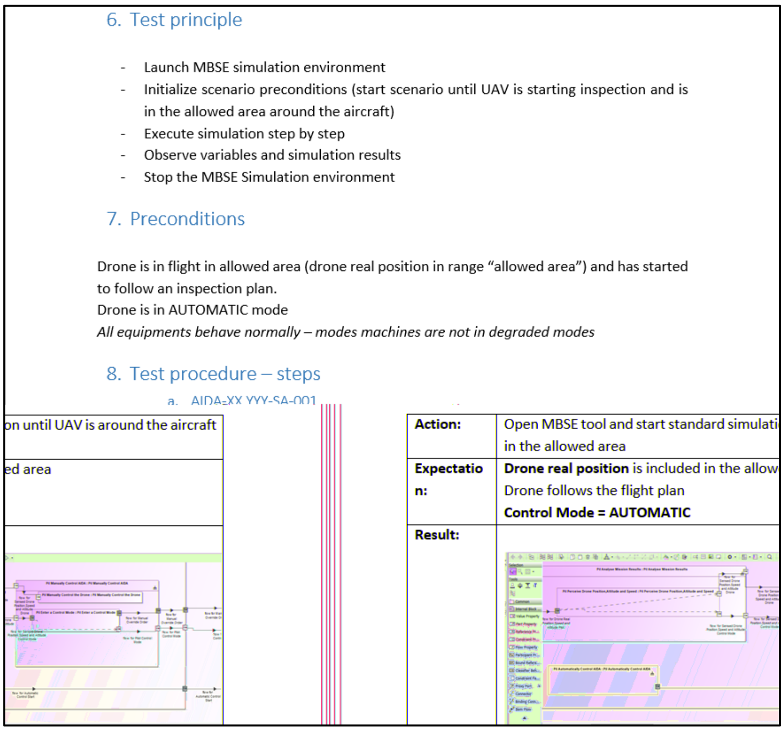
Scenario Concept	SE		SA		Value's domain correspondance	
	Variable/observer SE	SE Values	Variable/observer SA	SA Values	SE	SA
Drone in flighy	Drone real position	XYZ position vector value included in specific shape	N/A: Voir commentaires	-	-	-
Drone insied authorized area	Drone real position	XYZ position vector value included in specific shape	N/A: Voir commentaires	-	-	-
Automatic Mode	Control Mode Internal	AUTOMATIC Flight Plan AUTOMATIC Speed Consign	Sortie SF4.3.2	AUTO	AUTOMATIC Flight Plan	AUTO
					AUTOMATIC Speed Consign	
	Control Mode Pilote		EXT_PilotDetection.PilotSelectedMode		MANUAL	MANUAL
Loss of motor	Motor x motion	0	SF1i2_CreateMotion	LOST (failure mode "fail_loss")	0	LOST
					!=0	OK / ERRONEOUS
Motor loss detection	Motor x runaway	TRUE	SF7.SF3_MonitorParameters.SF734_MonitorDroneMotors.Motro1Runaway	TRUE	TRUE	TRUE
					FALSE	FALSE
Shut off motor lost	Motor x disabled	TRUE	SF7.SF73.MonitorParameters.MotorsDisabled_MotorX	TRUE	TRUE/FALSE	TRUE/FALSE
Consequence motor loss	Global thrust	! = Required Thrust +/- acceptance margin	SF1.GlobalThrustAndTorque	ERRONEOUS	=Required Thrust +/- acceptance margin	OK
					0	LOST
					!=Required Thrust +/- acceptance margin AND !=0	ERRONEOUS
Pilot detection abnormal behavior	Sensed drone position	XYZ position vector value à comparer avec flight plan	SF2.DetectControlDronePosition	ERRONEOUS	= flight plan +/- acceptance margin	OK
					0	LOST
					!= flight plan +/- acceptance margin	ERRONEOUS
Manual Mode	Control Mode Internal	MANUAL Flight Plan MANUAL Speed Consign	Sortie SF4.3.2	MANUEL	AUTOMATIC Flight Plan	AUTO
					AUTOMATIC Speed Consign	
	Control Mode Pilote		EXT_PilotDetection.PilotSelectedMode		MANUAL Flight Plan	MANUEL
					MANUAL Speed Consign	
Loss of drone control	Global thrust	! = Required Thrust +/- acceptance margin	SF1.GlobalThrustAndTorque	ERRONEOUS	=Required Thrust +/- acceptance margin	OK
					0	LOST
					!=Required Thrust +/- acceptance margin AND !=0	ERRONEOUS
Propeller off	Propeller x thrust	0	SF1.SF11_ControlHelix1.SF1i3_CreateThrustAndTorque.output	LOST	=Required Propeller Thrust +/- acceptance margin	OK
					0	LOST
					!=Required Propeller Thrust +/- acceptance margin AND !=0	ERRONEOUS
Motor shutdown power	Motor Shutdown	TRUE	SF1.SF11_ControlHelix1.SF1i6_DepowerMotor	TRUE	TRUE	TRUE
					FALSE	FALSE

SSR method

5.1.5 Common activity 3 – Transformation from Scenario to Procedure

Transformation from Scenario to Procedure

(a) Example of SE Verification procedure



6. Test principle

- Launch MBSE simulation environment
- Initialize scenario preconditions (start scenario until UAV is starting inspection and is in the allowed area around the aircraft)
- Execute simulation step by step
- Observe variables and simulation results
- Stop the MBSE Simulation environment

7. Preconditions

Drone is in flight in allowed area (drone real position in range "allowed area") and has started to follow an inspection plan.
Drone is in AUTOMATIC mode
All equipments behave normally – modes machines are not in degraded modes

8. Test procedure – steps

a. AIDA-XX YYY-SA-001

on until UAV is around the aircraft	
ed area	

Action:	Open MBSE tool and start standard simulati in the allowed area
Expectation:	**Drone real position** is included in the allow Drone follows the flight plan **Control Mode = AUTOMATIC**
Result:	

(b) Example of SA Verification procedure

This is not a current SA practise to find such writing test activities

11. Test procedure

a. AIDA--V443.001-SA-001

Action:	Look for function "SF1i2_CreateMotion" in your left menu and trigger its failure mode "fail_loss". Double-click on "fail_loss" failure mode to trigger it on the simulation. Failure modes are nested under the function in the *Model Explorer* left menu.
Expected results:	Failure mode "fail_loss" is triggered on the simulation. The failure mode once is triggered it should have disappeared on the menu. On the layout, failed components are displayed in red colour.
Observed result:	"fail_loss" failure mode does not appear any more for the function "SF1i2_CreateMotion". Many variables have changed its status after triggering the failure mode, some red colours are displayed in the main view.
Status:	OK
Note (optional):	N/A

b. AIDA--V443.001-SA-002

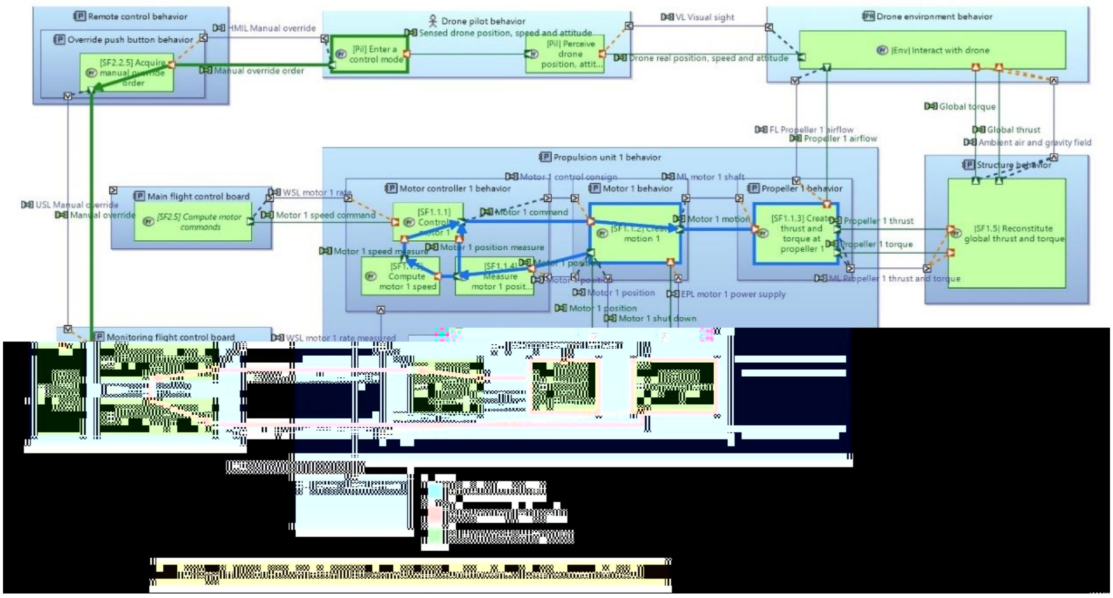
Action:	Check for observers output values considered in this test
Expected results:	• Motor 1 (failed motor) is disabled: SF7.SF73.MonitorParameters.MotorsDisabled_Motor1 = TRUE • Control mode is no longer AUTO, it is MANUAL now: EXT_PilotDetection.mode = AUTO • Drone loss of control FC01_CAT_Uncontrolled_drone_crash_in_unauthorized_area = TRUE FC02_HAZ_Uncontrolled_drone_crash_in_authorized_area = TRUE
Observed result:	Following results are obtained: SF7.SF73.MonitorParameters.MotorsDisabled_Motor1 = TRUE EXT_PilotDetection.mode = AUTO FC01_CAT_Uncontrolled_drone_crash_in_unauthorized_area = TRUE FC02_HAZ_Uncontrolled_drone_crash_in_authorized_area = TRUE
Status:	Not passed

5.1.6 PoC A - PoC SE model with scenarios (not executable) and SA executable model

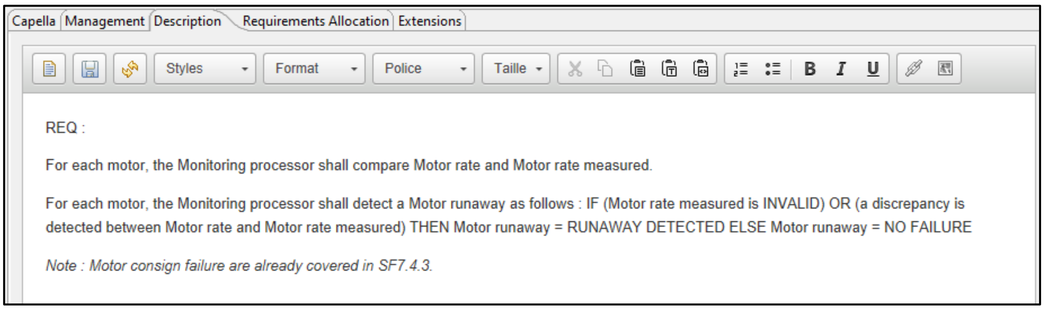
(a) Input Specialization

(i) SE Models:

- **Behaviors, Functions and associated Data Flows / Functional Chains:**



- **Behavior Textual Requirements for Functions regarding Validity evaluation:**



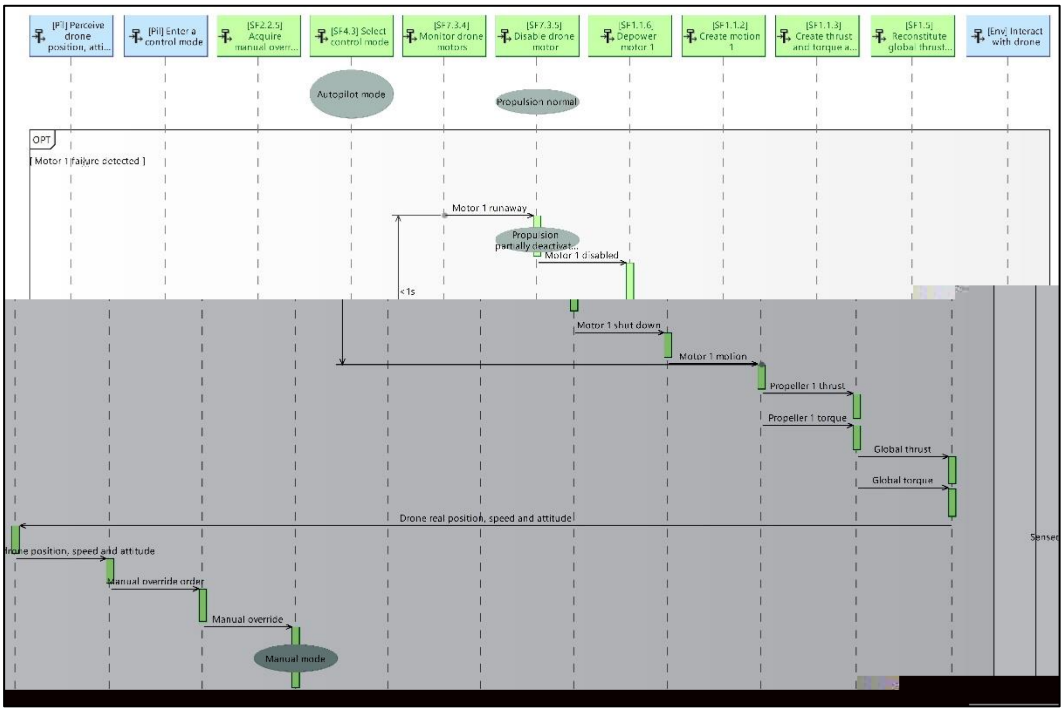
REQ :

For each motor, the Monitoring processor shall compare Motor rate and Motor rate measured.

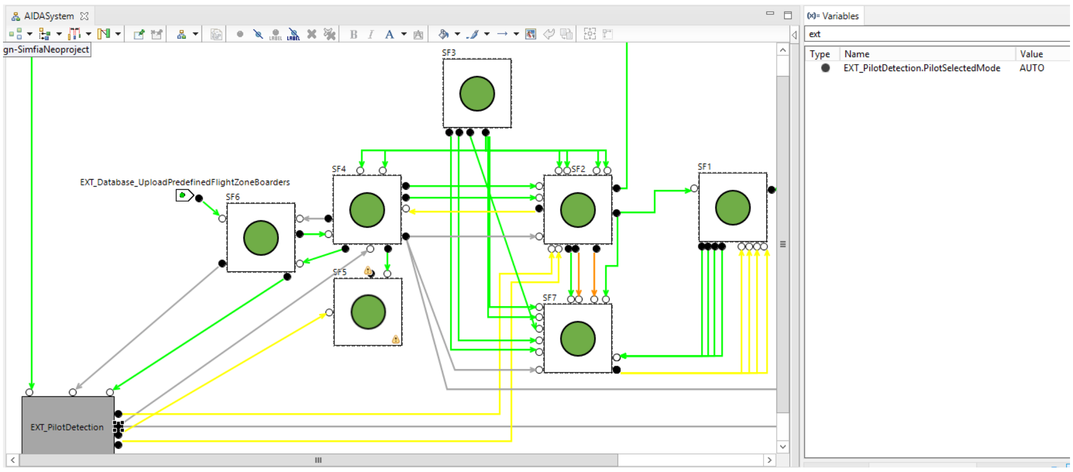
For each motor, the Monitoring processor shall detect a Motor runaway as follows : IF (Motor rate measured is INVALID) OR (a discrepancy is detected between Motor rate and Motor rate measured) THEN Motor runaway = RUNAWAY DETECTED ELSE Motor runaway = NO FAILURE

Note : Motor consign failure are already covered in SF7.4.3.

- **Functional Exchanges Scenarios:**



(ii) SA Models :

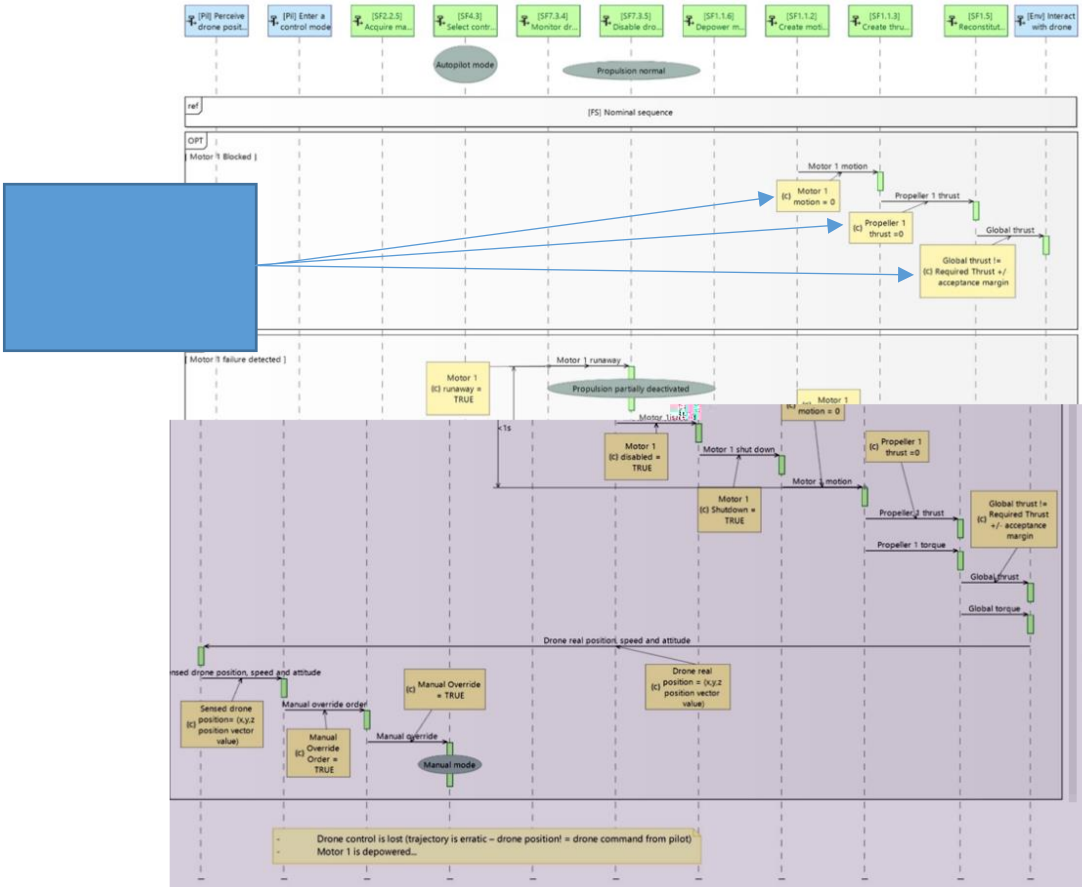


extended

(b) PoC Results

(i) SE

- _________: Motor Loss

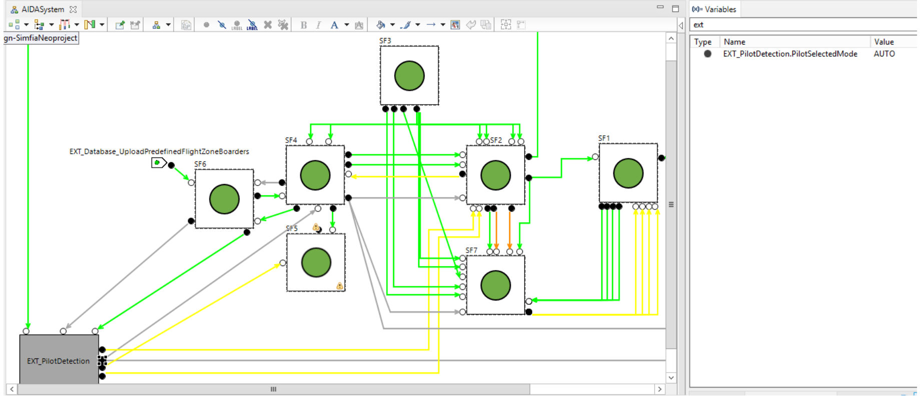


(ii) SA

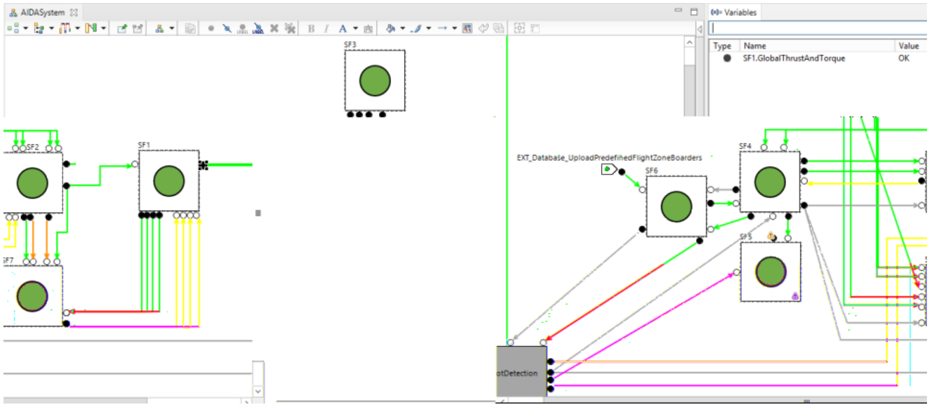
AIDA-XX.YYY-SA-001

Action:	
Expectation:	EXT_PilotDetection.PilotSelectedMode = AUTO
Result:	EXT_PilotDetection.PilotSelectedMode = AUTO

____

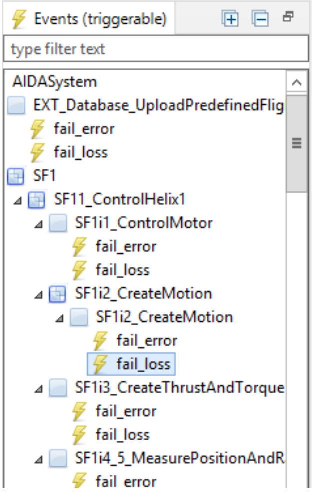
	
Status:	
Note (opt.):	

AIDA-XX.YYY-SA-002

Action:	
Expectation:	**SF1.GlobalThrustAndTorque = OK**
Result:	**SF1.GlobalThrustAndTorque = OK** 
Status:	
Note (opt.):	

AIDA-XX.YYY-SA-003

Action:	**SF1i2_CreateMotion** **fail_loss** **fail_loss**
Expectation:	

Result:	
Status:	
Note (opt.l):	

AIDA-XX.YYY-SA-004

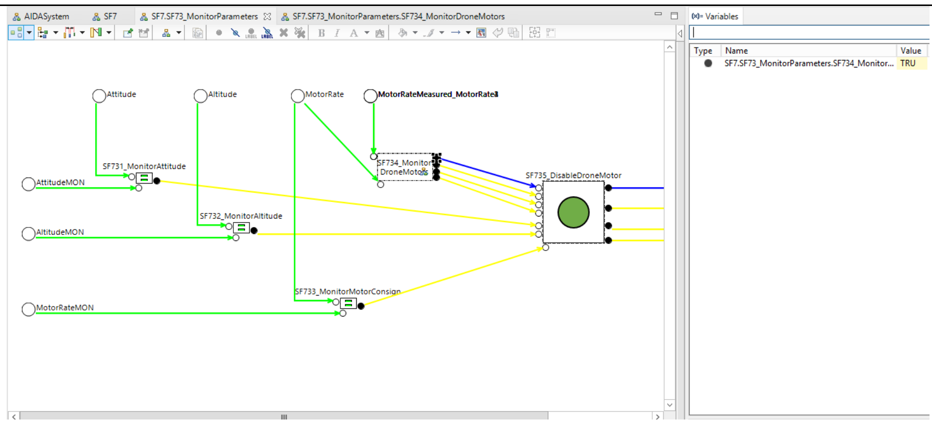
Action:	
Expectation:	**SF1.SF11_ControlHelix1.SF1i3_CreateThrustAndTorque.output = LOST**
Result:	**SF1.SF11_ControlHelix1.SF1i3_CreateThrustAndTorque.output = LOST**
Status:	
Note (opt.l):	

AIDA-XX.YYY-SA-005

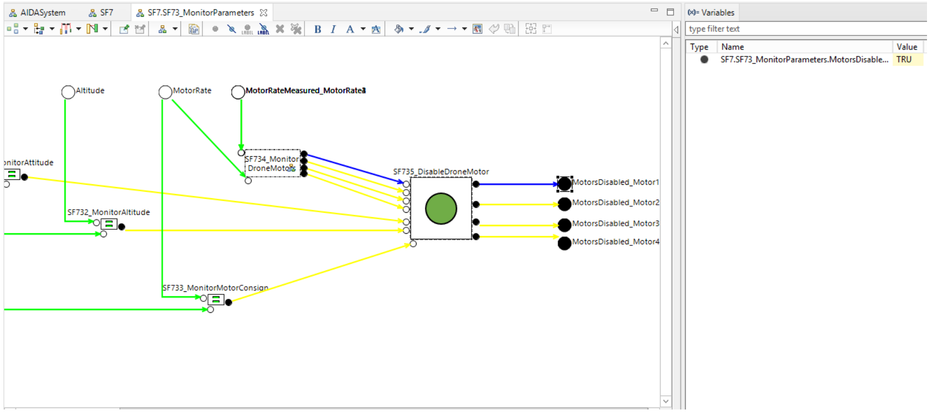
Action:	
Expectation:	**SF1.GlobalThrustAndTorque = ERRONEOUS**
Result:	**SF1.GlobalThrustAndTorque = ERRONEOUS**
Status:	
Note (opt.):	

AIDA-XX.YYY-SA-006

Action:	
Expectation:	**SF7.SF3_MonitorParameters.SF734_MonitorDroneMotors.Motro1Runaway = TRUE**
Result:	**SF7.SF3_MonitorParameters.SF734_MonitorDroneMotors.Motro1Runaway = TRUE**

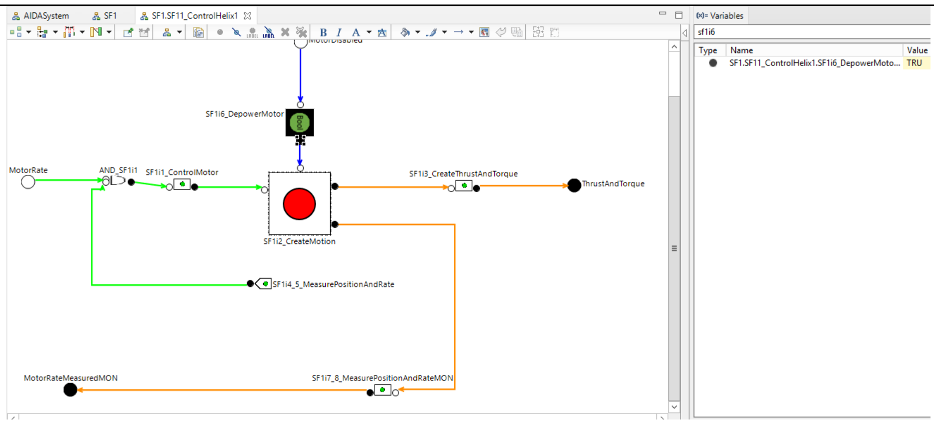
	
Status:	
Note (opt.):	

AIDA-XX.YYY-SA-007

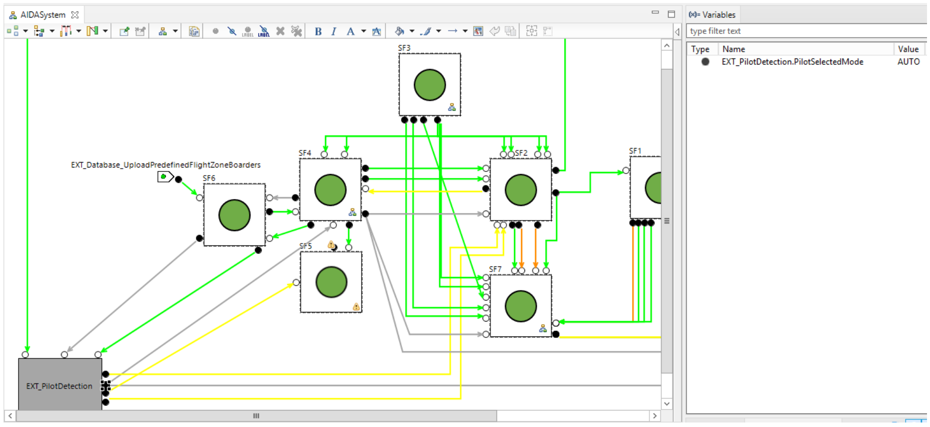
Action:	
Expectation:	**SF7.SF73.MonitorParameters.MotorsDisabled_MotorX = TRUE**
Result:	**SF7.SF73.MonitorParameters.MotorsDisabled_MotorX = TRUE** 
Status:	
Note (opt.):	

AIDA-XX.YYY-SA-008

Action:	
Expectation:	**SF1.SF11_ControlHelix1.SF1i6_DepowerMotor = TRUE**
Result:	**SF1.SF11_ControlHelix1.SF1i6_DepowerMotor = TRUE**

	
Status:	
Note (opt.):	

AIDA-XX.YYY-SA-009

	SA
Action:	
Expectation:	EXT_PilotDetection.PilotSelectedMode = MANUAL
Result:	EXT_PilotDetection.PilotSelectedMode = AUTO 
Status:	
Note (opt.):	

(c) PoC Conclusion

5.1.7 PoC B - PoC executable SE model

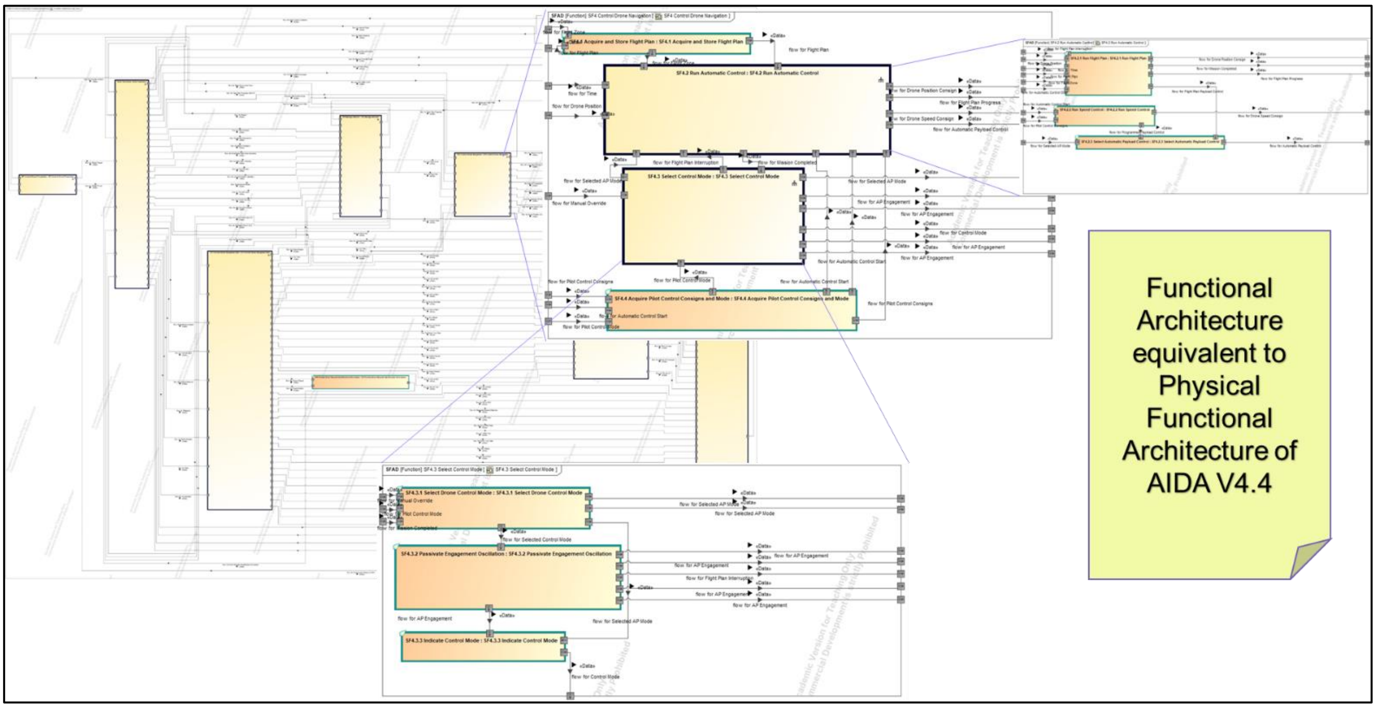
(a) PoC Specialization

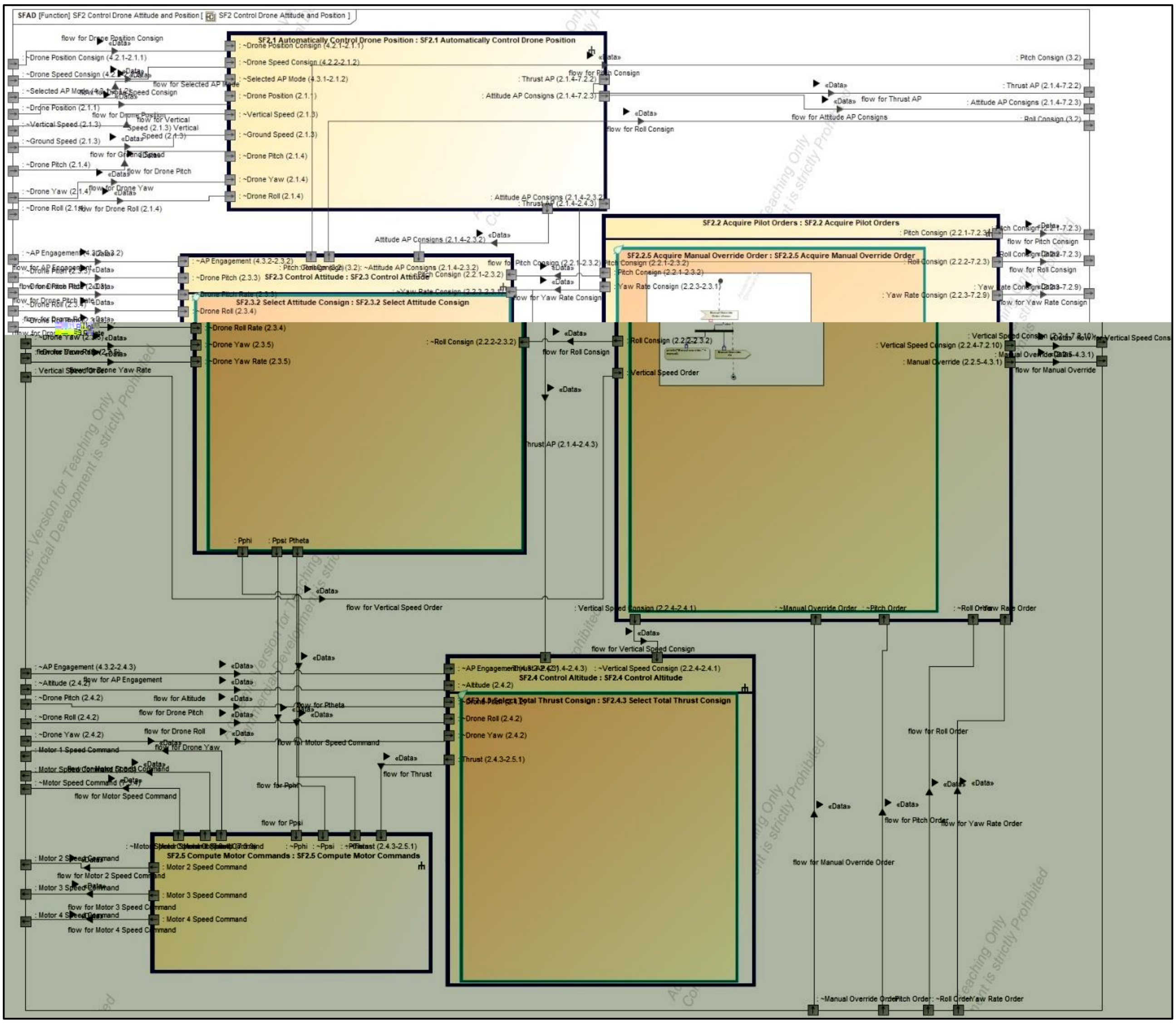
PoC A ***PoC B***

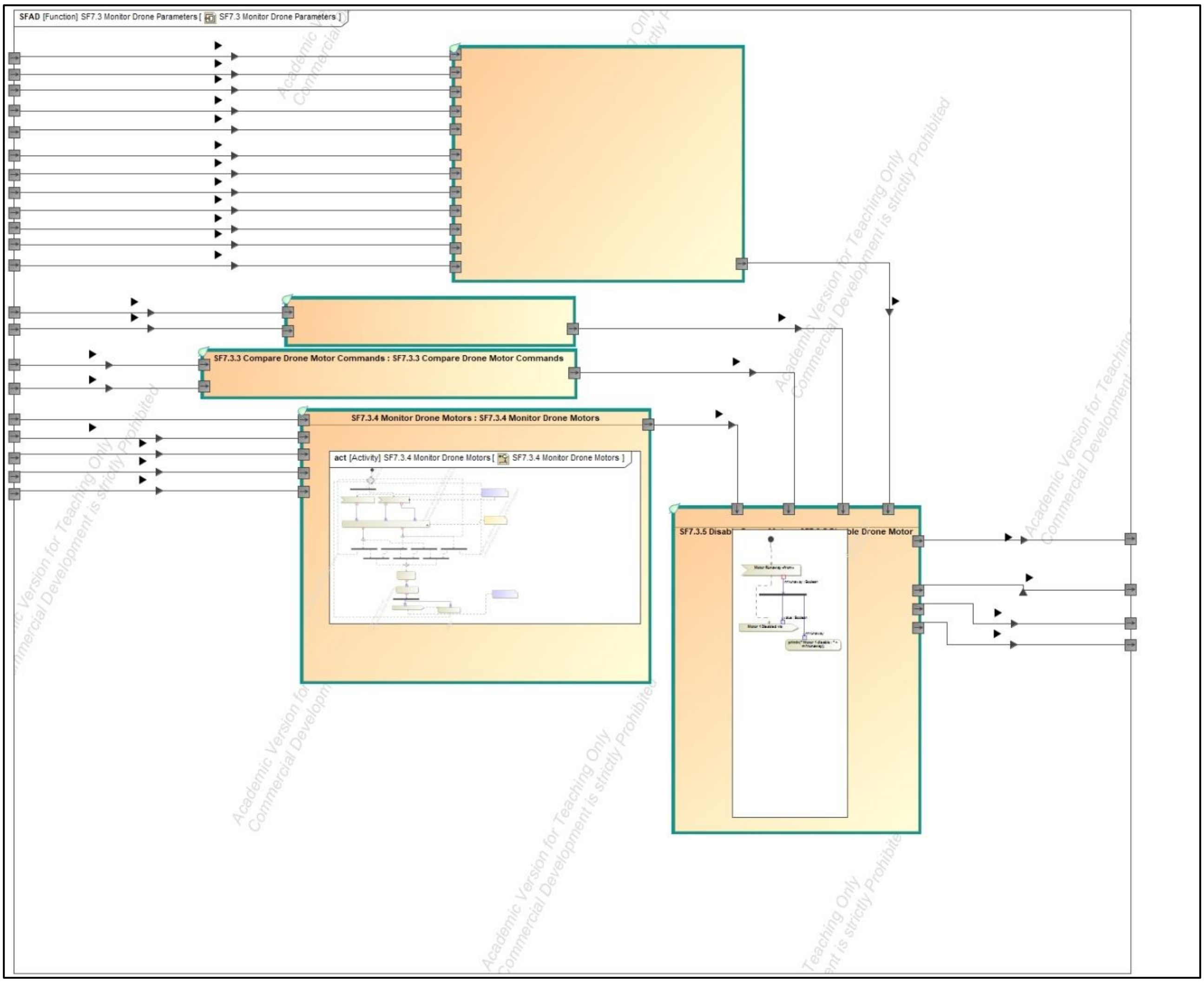
Annex *SAMAREQ Profile*

Annex *Derived Result*

➢ **SysML model – Functional Architecture overview**



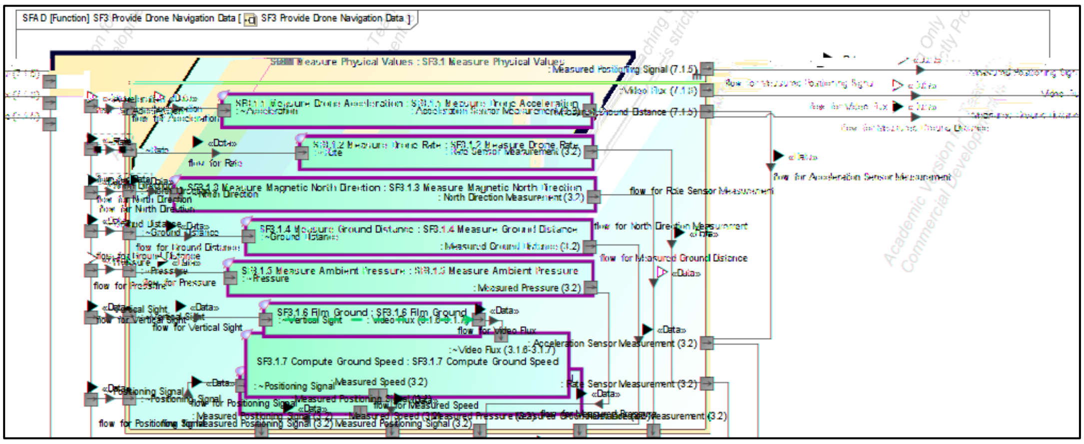

SFAD [Function] SF2 Control Drone Attitude and Position [SF2 Control Drone Attitude and Position]
SF2.1 Automatically Control Drone Position : SF2.1 Automatically Control Drone Position
SF2.2 Acquire Pilot Orders : SF2.2 Acquire Pilot Orders
SF2.2.5 Acquire Manual Override Order : SF2.2.5 Acquire Manual Override Order
SF2.3 Control Attitude : SF2.3 Control Attitude
SF2.3.2 Select Attitude Consign : SF2.3.2 Select Attitude Consign
SF2.4 Control Altitude : SF2.4 Control Altitude
SF2.4.3 Select Total Thrust Consign : SF2.4.3 Select Total Thrust Consign
SF2.5 Compute Motor Commands : SF2.5 Compute Motor Commands



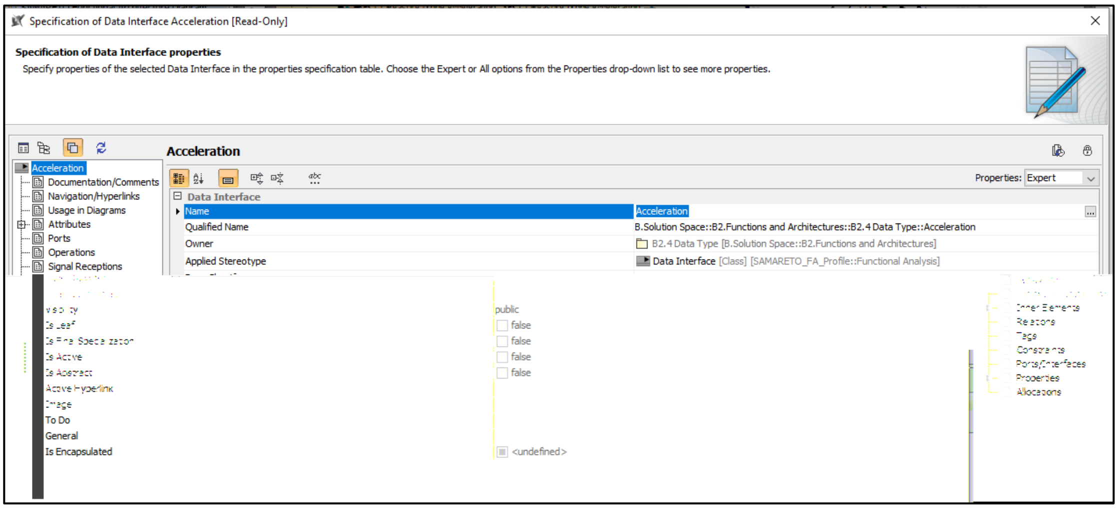
- **<u>Behavior associated to Leaf Function</u>**

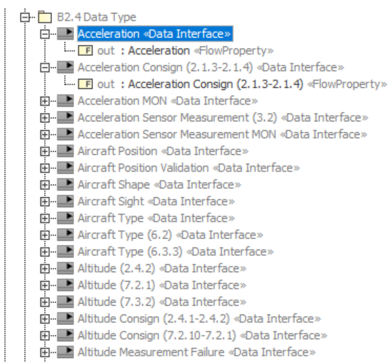
- **<u>Generalization of delegation mechanism</u>**

➢ Interface and Data Flows Specification

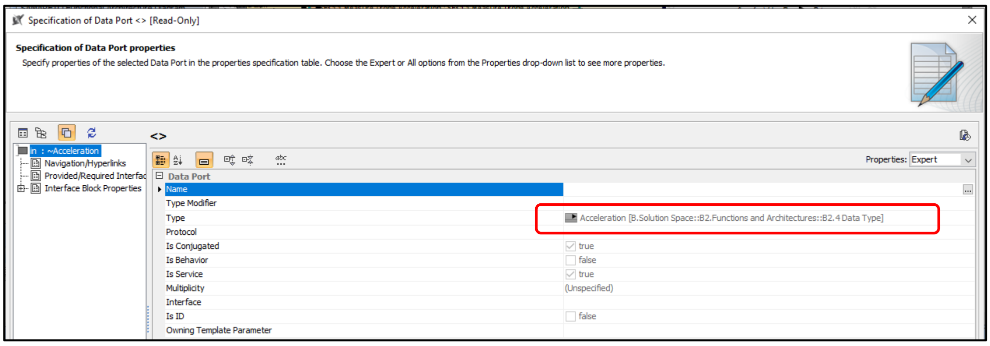


○ Interface Block Type :

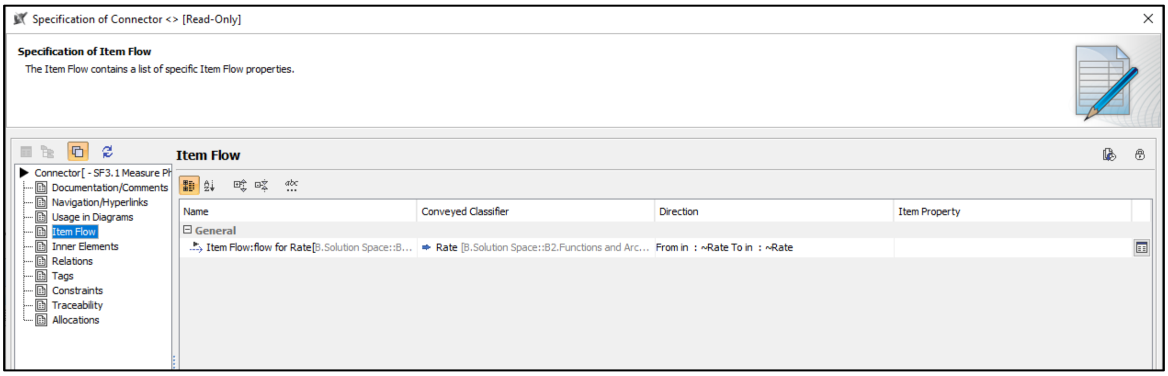


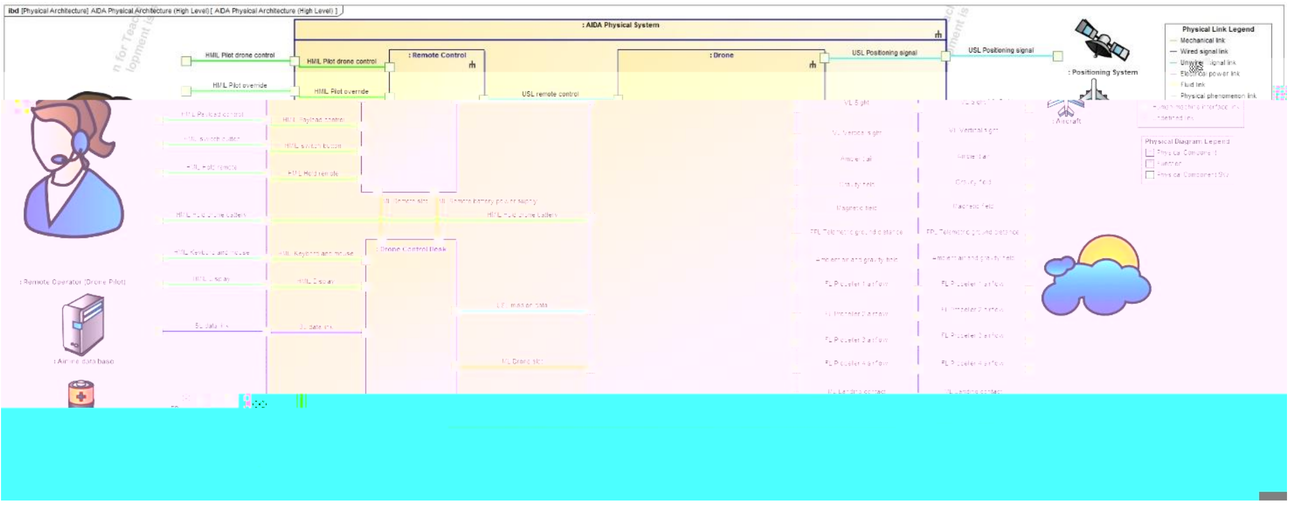


- ## Port Definition



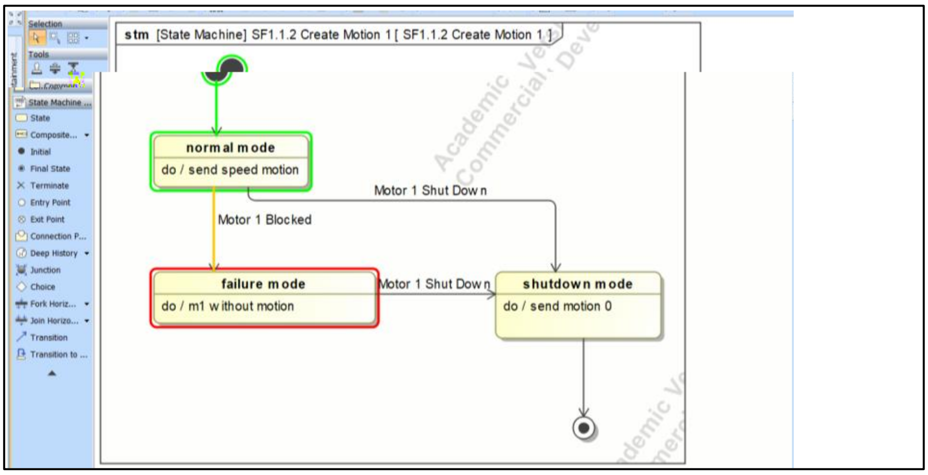
- ## Item Flows

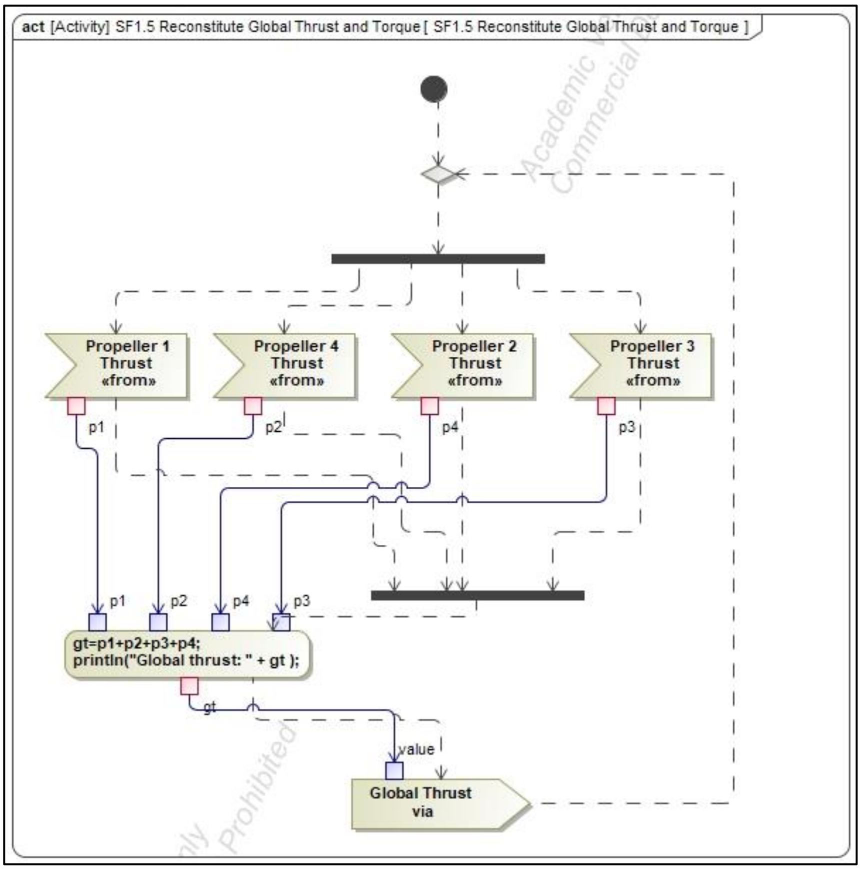


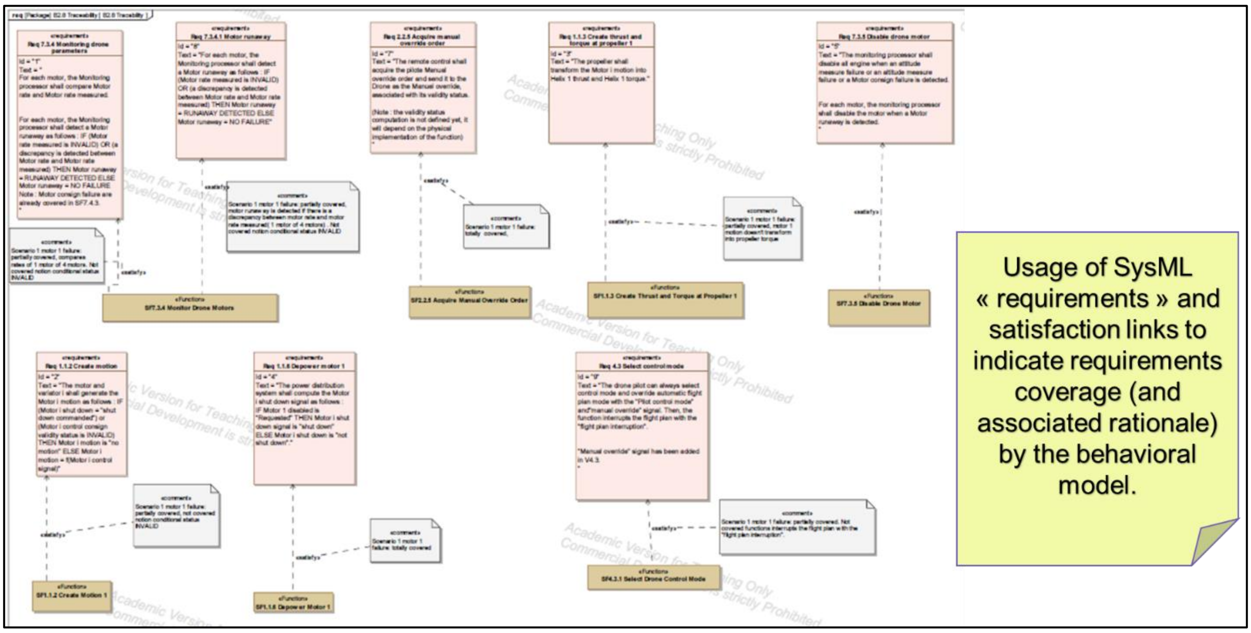


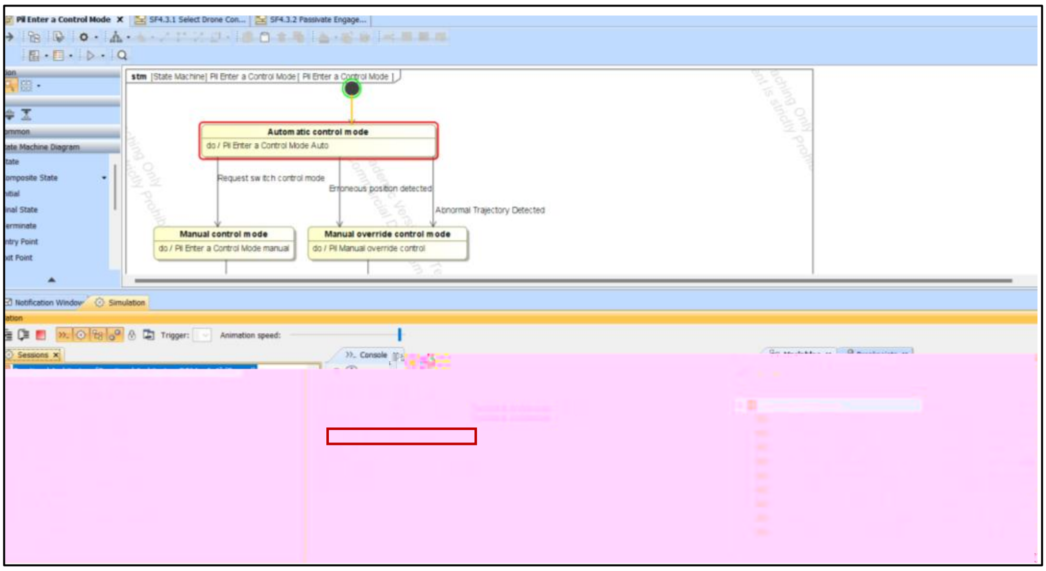
➢ SysML model – Behavioral overview

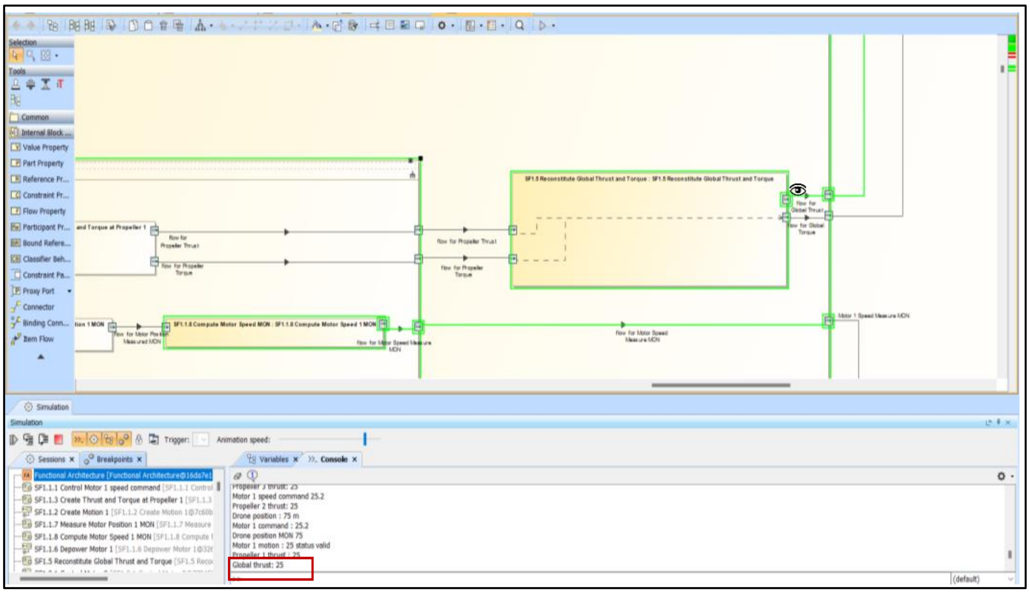
Loss of a Motor

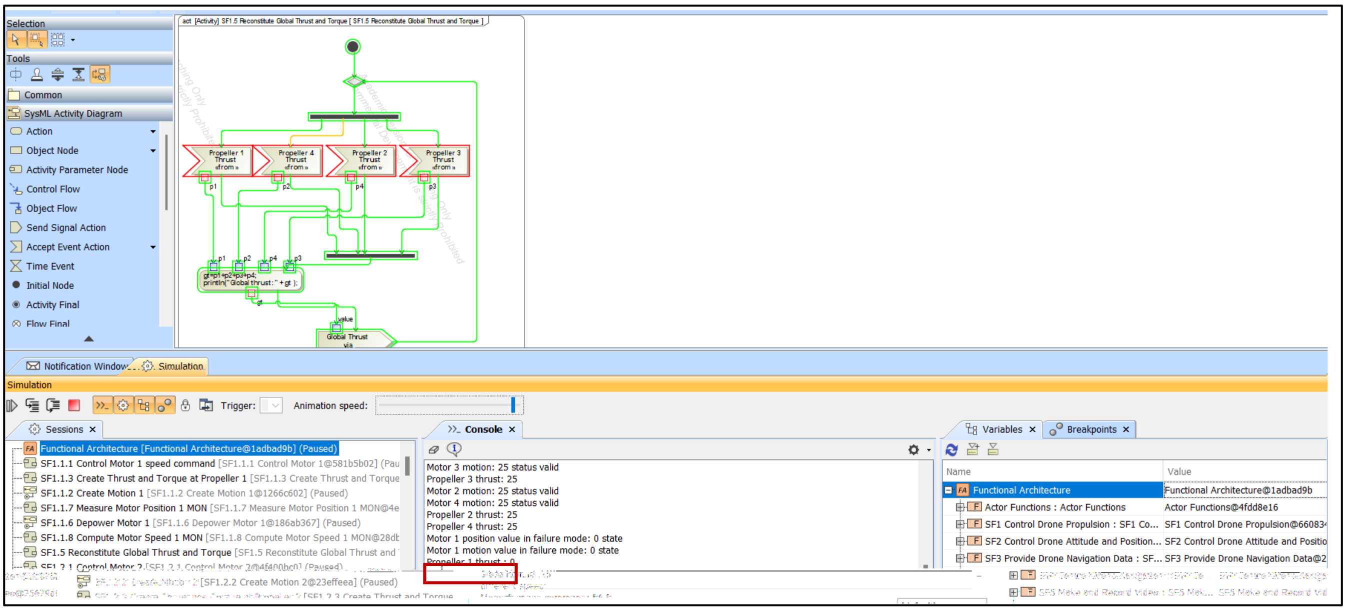


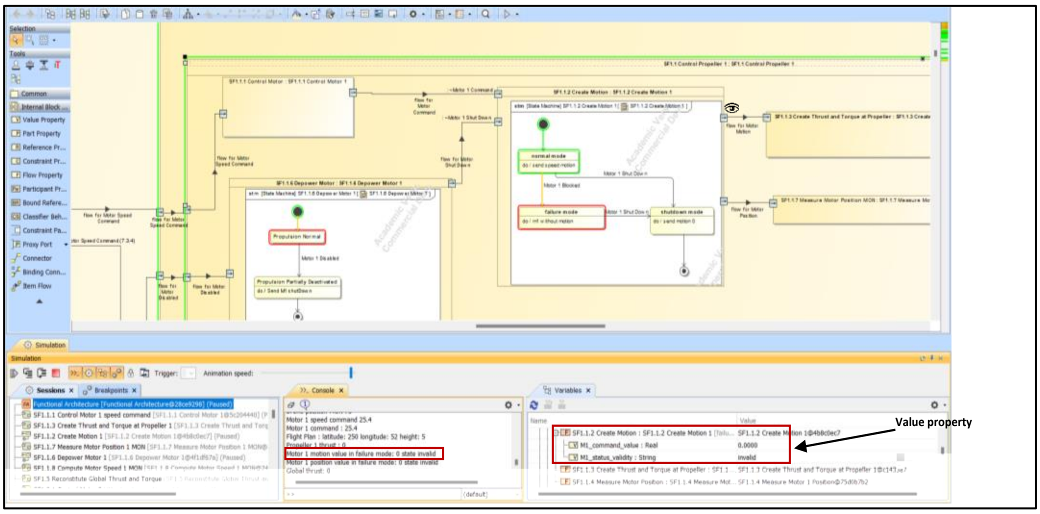

act [Activity] SF1.5 Reconstitute Global Thrust and Torque [SF1.5 Reconstitute Global Thrust and Torque]
Propeller 1 Thrust «from»
Propeller 4 Thrust «from»
Propeller 2 Thrust «from»
Propeller 3 Thrust «from»
p1
p2
p4
p3
gt=p1+p2+p3+p4;
println("Global thrust: " + gt);
gt
value
Global Thrust via

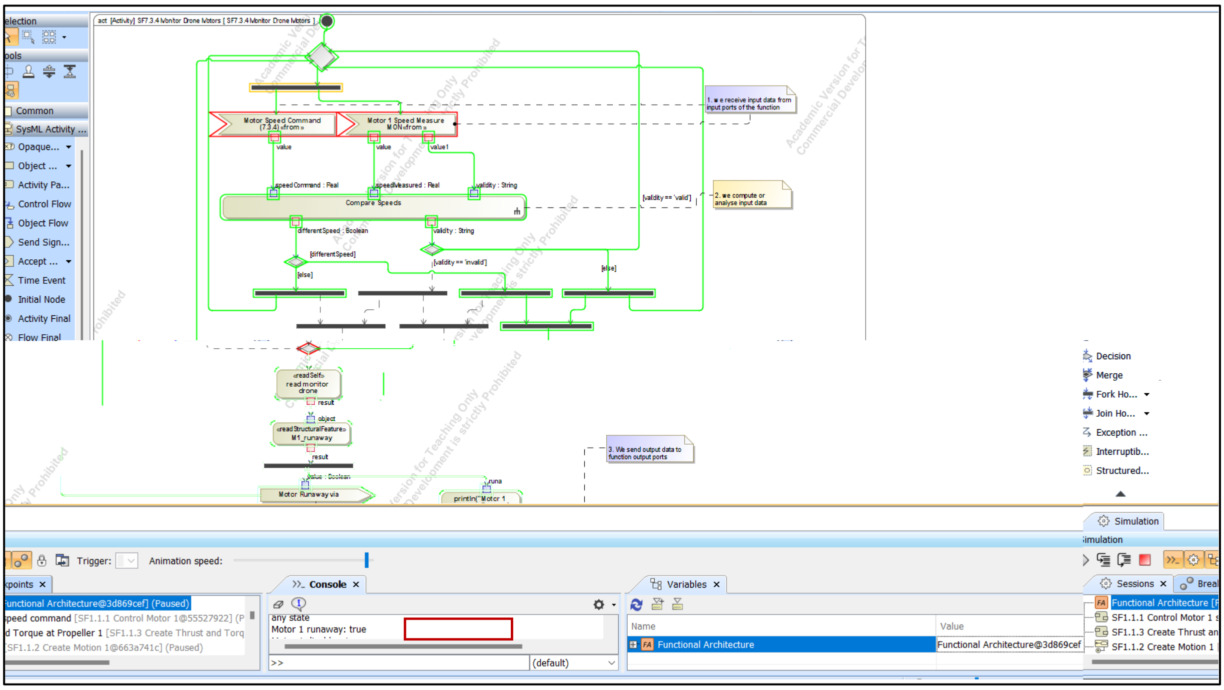

Usage of SysML « requirements » and satisfaction links to indicate requirements coverage (and associated rationale) by the behavioral model.


Automatic control mode
do / PI Enter a Control Mode Auto
Request switch control mode
Erroneous position detected
Abnormal Trajectory Detected
Manual control mode
do / PI Enter a Control Mode manual
Manual override control mode
do / PI Manual override control


SF1.5 Reconstitute Global Thrust and Torque : SF1.5 Reconstitute Global Thrust and Torque
SF1.1.8 Compute Motor Speed MON : SF1.1.8 Compute Motor Speed 1 MON
Motor 1 speed command 25.2
Propeller 2 thrust: 25
Drone position : 75 m
Motor 1 command : 25.2
Drone position MON 75
Motor 1 motion : 25 status valid
Global thrust: 25

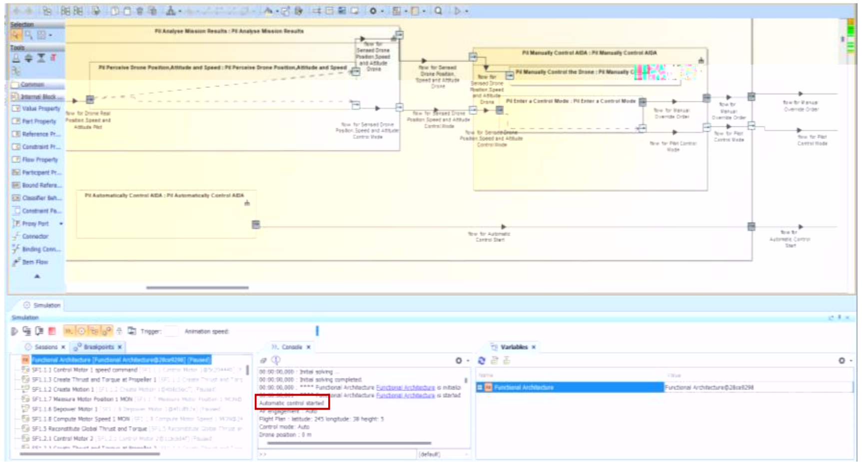
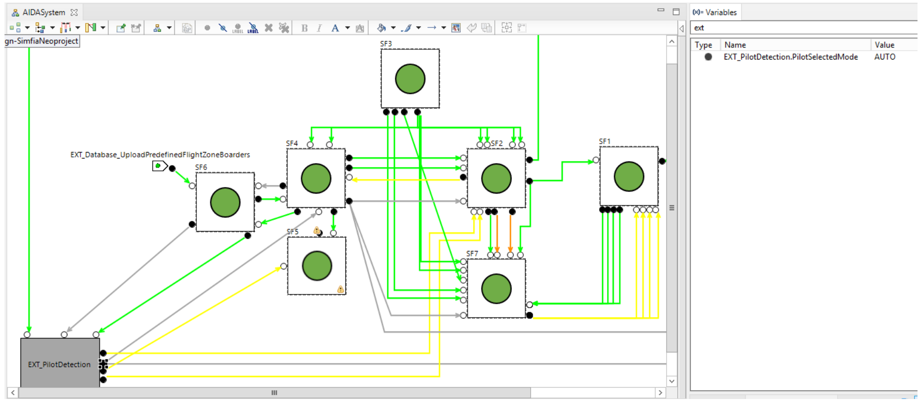

Selection
Tools
Common
SysML Activity Diagram
Action
Object Node
Activity Parameter Node
Control Flow
Object Flow
Send Signal Action
Accept Event Action
Time Event
Initial Node
Activity Final
Flow Final
Propeller 1 Thrust
Propeller 4 Thrust
Propeller 2 Thrust
Propeller 3 Thrust
Global Thrust
Notification Window
Simulation
Trigger:
Animation speed:
Sessions
Functional Architecture [Functional Architecture@1adbad9b] (Paused)
SF1.1.1 Control Motor 1 speed command [SF1.1.1 Control Motor 1@581b5b02] (Pau
SF1.1.3 Create Thrust and Torque at Propeller 1 [SF1.1.3 Create Thrust and Torque
SF1.1.2 Create Motion 1 [SF1.1.2 Create Motion 1@1266c602] (Paused)
SF1.1.7 Measure Motor Position 1 MON [SF1.1.7 Measure Motor Position 1 MON@4e
SF1.1.6 Depower Motor 1 [SF1.1.6 Depower Motor 1@186ab367] (Paused)
SF1.1.8 Compute Motor Speed 1 MON [SF1.1.8 Compute Motor Speed 1 MON@28db
SF1.5 Reconstitute Global Thrust and Torque [SF1.5 Reconstitute Global Thrust and
Console
Motor 3 motion: 25 status valid
Propeller 3 thrust: 25
Motor 2 motion: 25 status valid
Motor 4 motion: 25 status valid
Propeller 2 thrust: 25
Propeller 4 thrust: 25
Motor 1 position value in failure mode: 0 state
Motor 1 motion value in failure mode: 0 state
Variables
Breakpoints
Name
Value
Functional Architecture
Functional Architecture@1adbad9b
Actor Functions : Actor Functions
Actor Functions@4fdd8e16
SF1 Control Drone Propulsion : SF1 Co...
SF2 Control Drone Attitude and Position...
SF3 Provide Drone Navigation Data : SF...

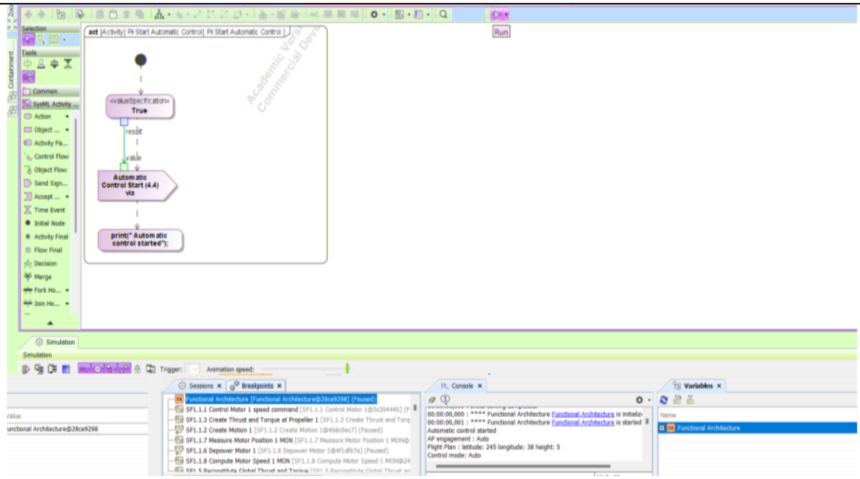

Value property


Common
SysML Activity ...
Opaque...
Object ...
Activity Pa...
Control Flow
Object Flow
Send Sign...
Accept ...
Time Event
Initial Node
Activity Final
Flow Final
Decision
Merge
Fork Ho...
Join Ho...
Exception ...
Interruptib...
Structured...
Simulation
Trigger:
Animation speed:
Console
Variables
Sessions
Name
Value
Functional Architecture
Functional Architecture@3d869cef
Motor 1 runaway: true
(default)
SF1.1.1 Control Motor 1
SF1.1.3 Create Thrust and Torq
SF1.1.2 Create Motion 1@663a741c] (Paused)

(b) PoC Results

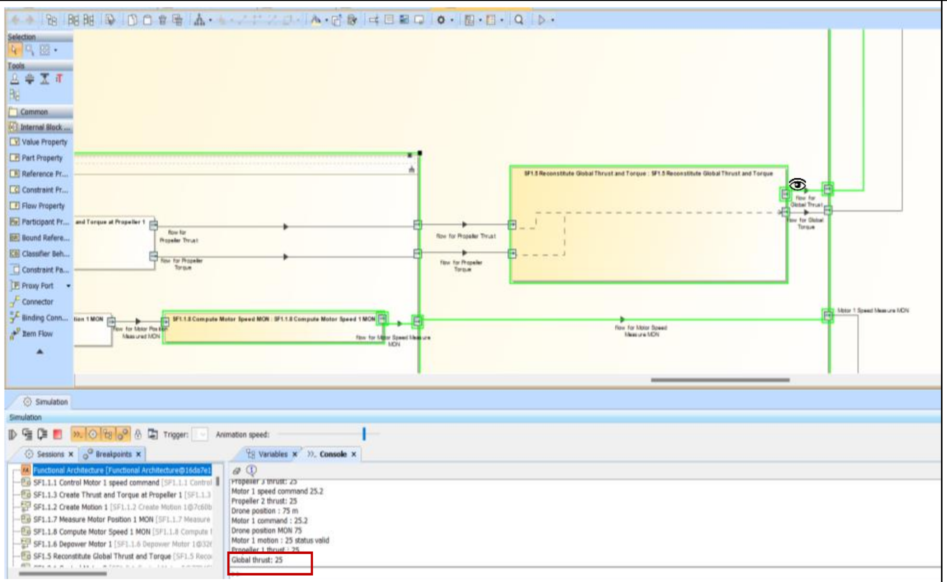
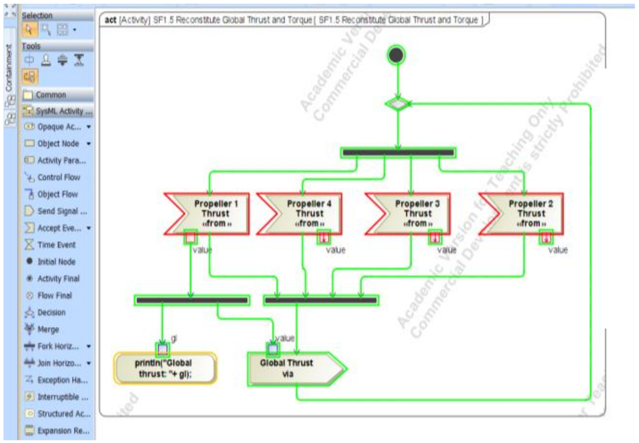
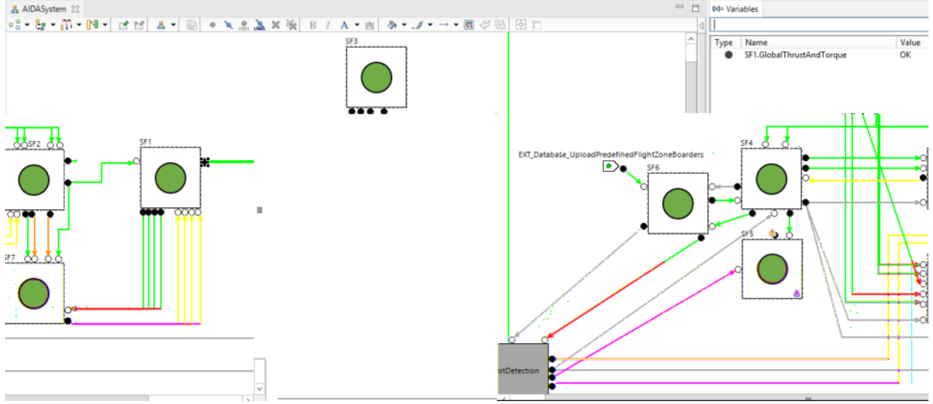
(i) AIDA-XX.YYY-SA-001

	SE	SA
Action:		
Expectation:	**Drone real position** **Control Mode = AUTOMATIC**	**EXT_PilotDetection.PilotSelectedMode = AUTO**
Result:		**EXT_PilotDetection.PilotSelectedMode = AUTO** 

		
Status:		
Note (opt.):		

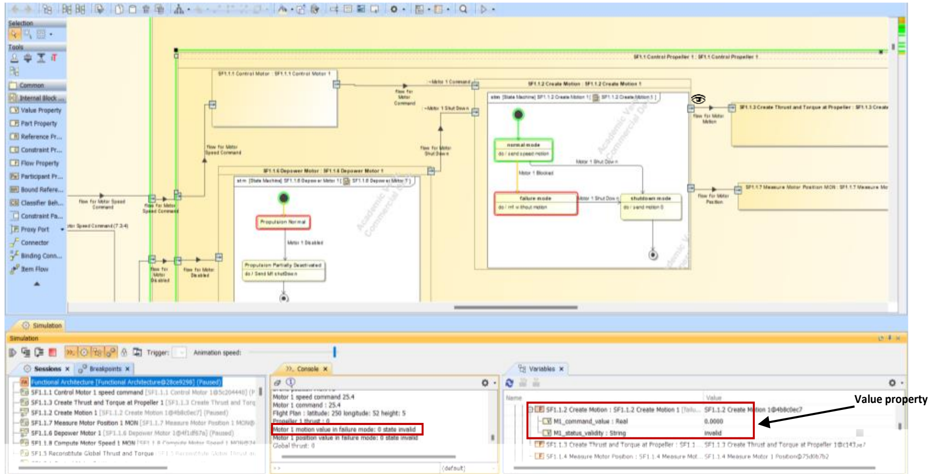
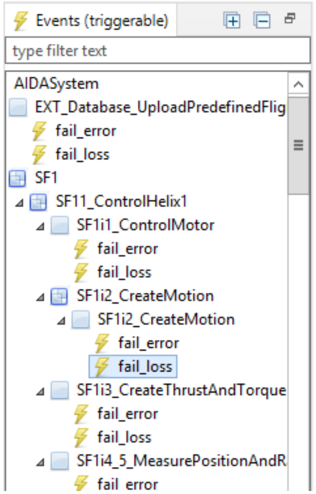
(ii) AIDA-XX.YYY-SA-002

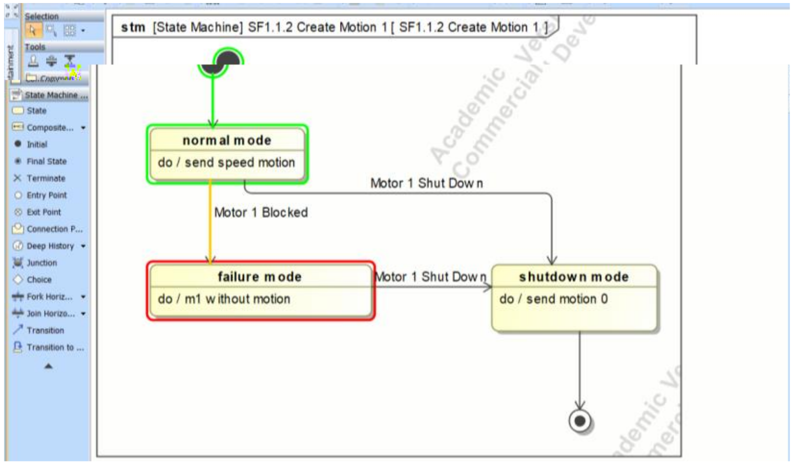
	SE	SA
Action:		
Expectation:	Global Thrust = Required thrust +/- acceptance margin	SF1.GlobalThrustAndTorque = OK
Result:		SF1.GlobalThrustAndTorque = OK

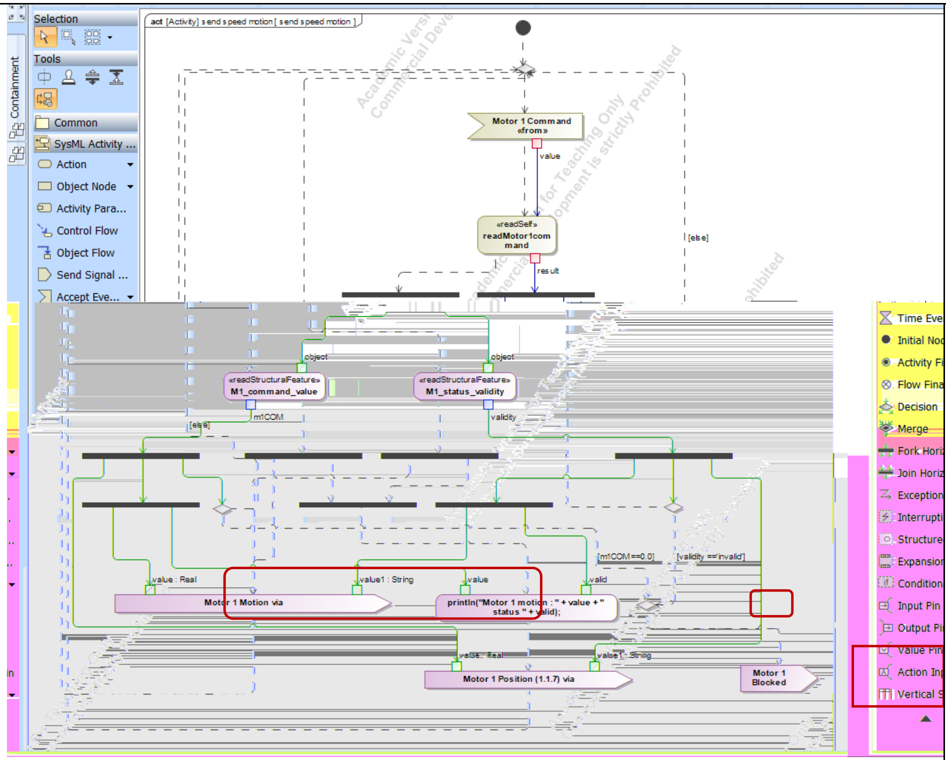
<table>
<tr><td></td><td>
</td><td></td></tr>
<tr><td>Status:</td><td></td><td></td></tr>
</table>

Note (opt.):		

(iii) AIDA-XX.YYY-SA-003

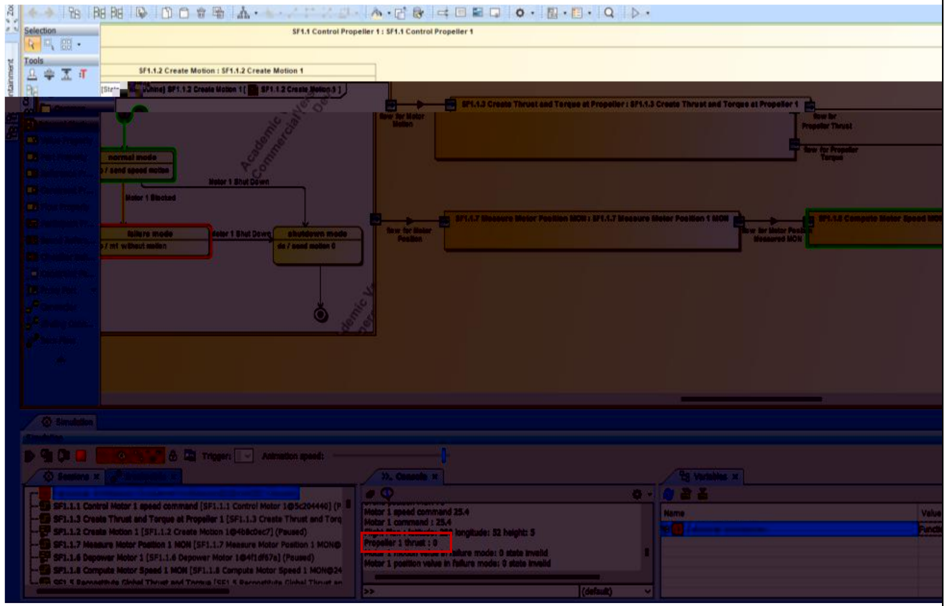
	SE	SA
Action:		SF1i2_CreateMotion fail_loss fail_loss
Expectation:	*Motor x motion = 0*	
Result:		

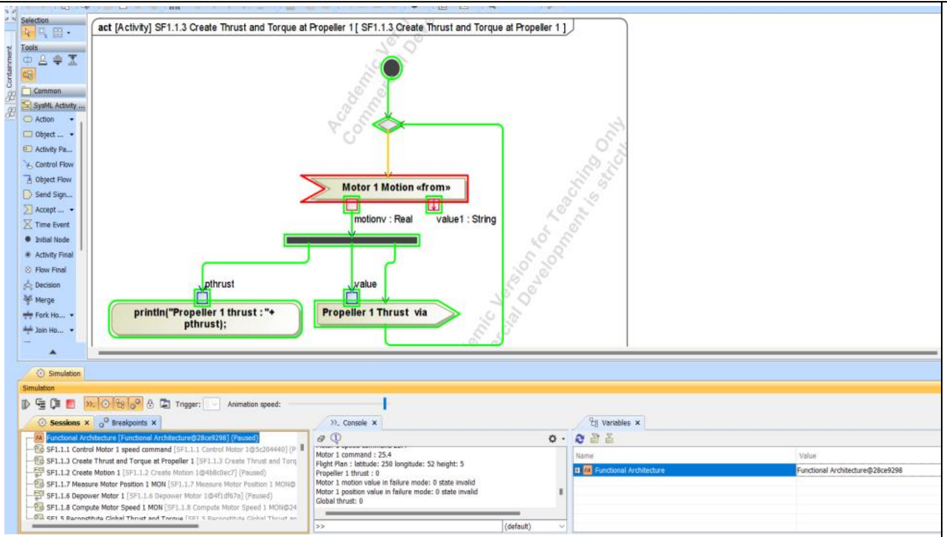
	stm [State Machine] SF1.1.2 Create Motion 1 [SF1.1.2 Create Motion 1] normal mode do / send speed motion Motor 1 Shut Down Motor 1 Blocked failure mode do / m1 without motion Motor 1 Shut Down shutdown mode do / send motion 0 State Initial Final State Terminate Entry Point Exit Point Junction Choice Transition	

		
Status:		
Note (opt.l):		

(iv) AIDA-XX.YYY-SA-004

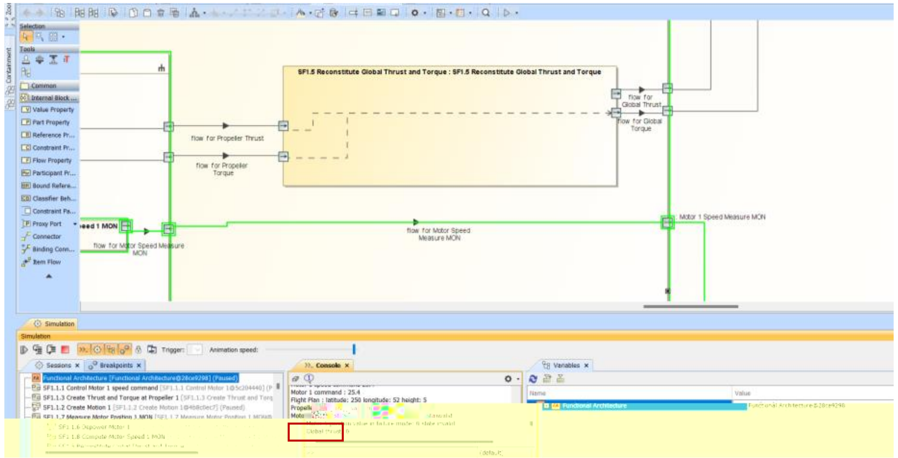
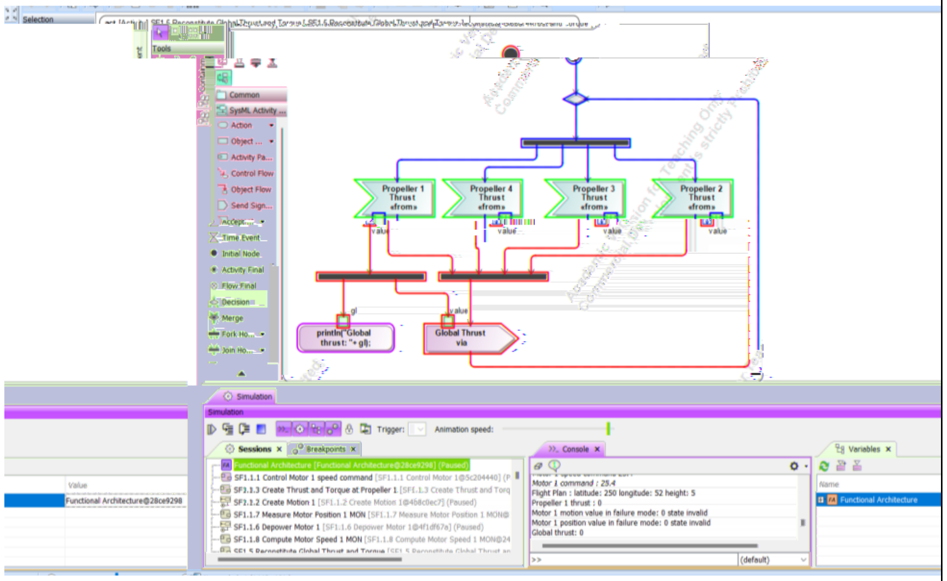
	SE	SA
Action:		
Expectation:	***Propeller x thrust = 0***	**SF1.SF11_ControlHelix1.SF1i3_CreateThrustAndTorque.output = LOST**

Result:	SF1.1 Control Propeller 1 : SF1.1 Control Propeller 1 SF1.1.2 Create Motion : SF1.1.2 Create Motion 1 normal mode failure mode shutdown mode Propeller 1 thrust : 0	

		
Status:		
Note (opt.l):		

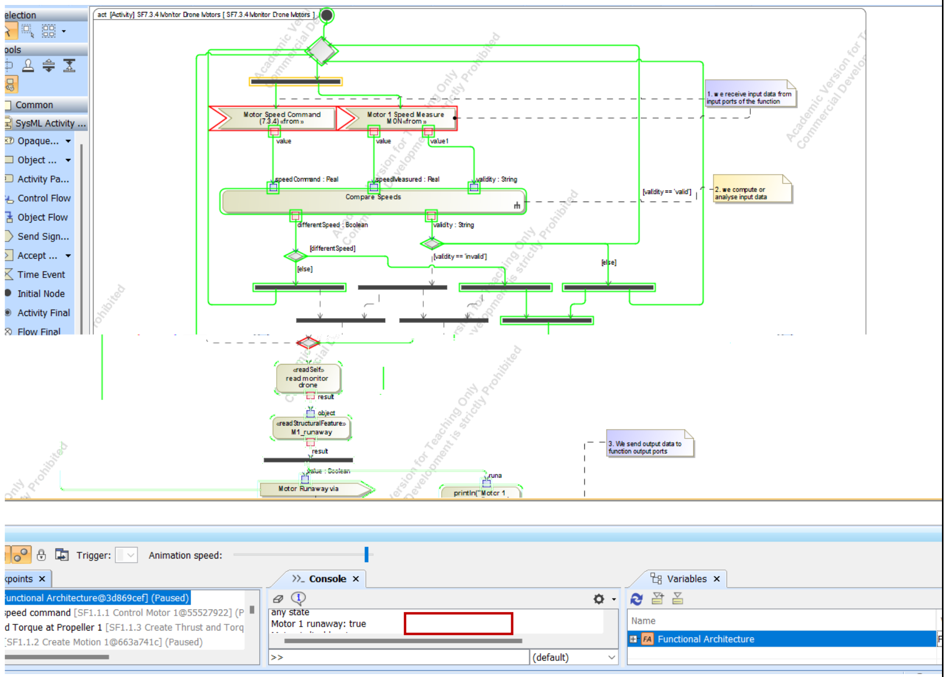
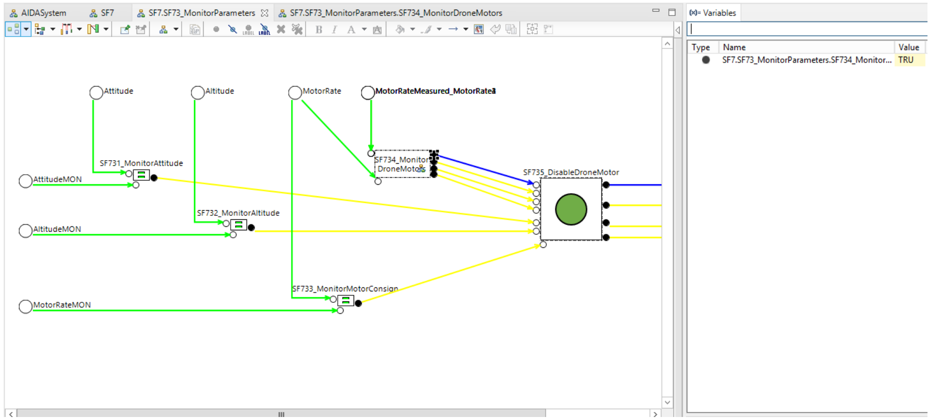
(v) AIDA-XX.YYY-SA-005

	SE	SA
Action:		
Expectation:	Global thrust! = Required Thrust +/- acceptance margin	SF1.GlobalThrustAndTorque = ERRONEOUS

Result:	 	**SF1.GlobalThrustAndTorque = ERRONEOUS**

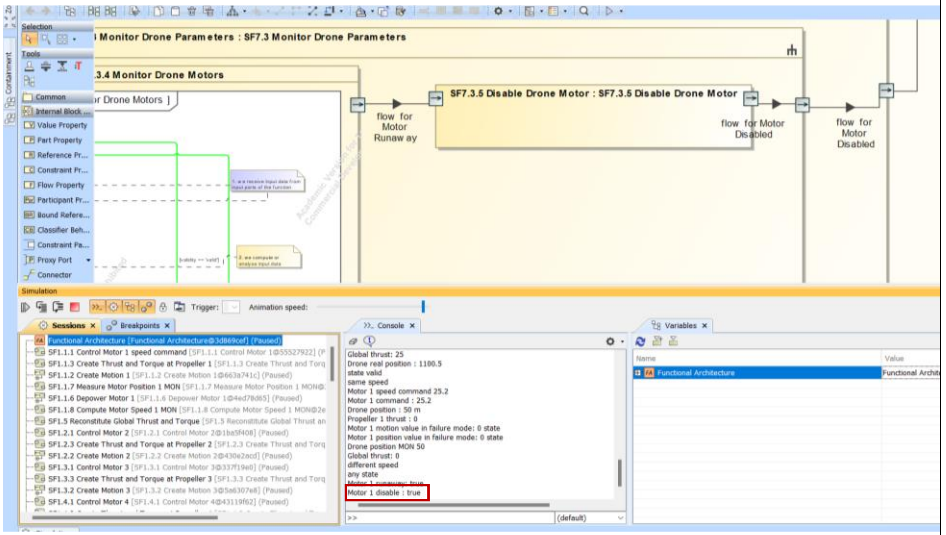
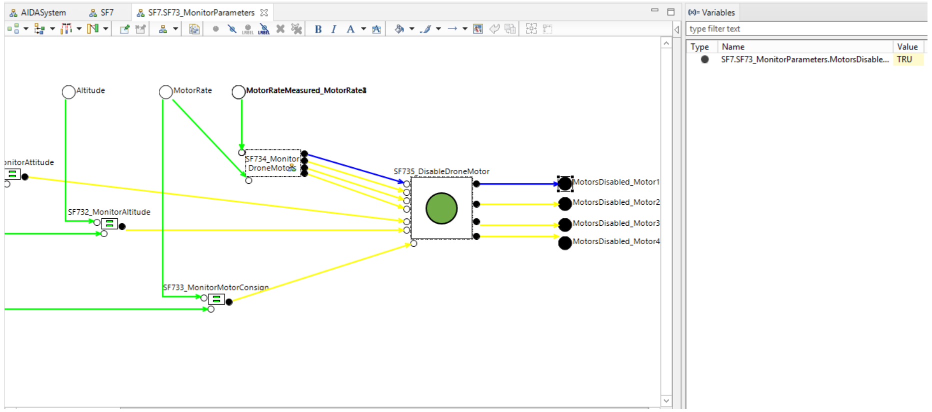
Status:		
Note (opt.):		

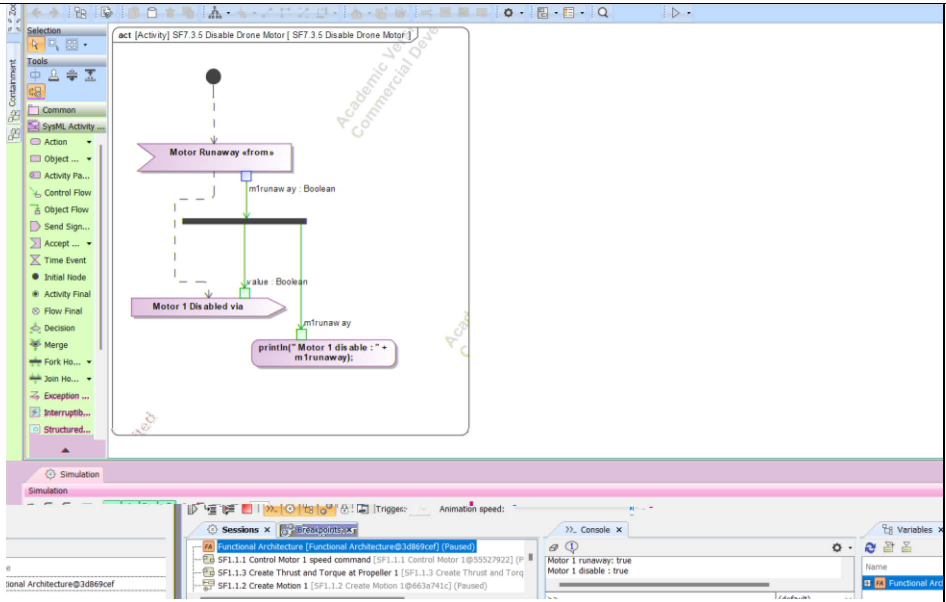
(vi) AIDA-XX.YYY-SA-006

	SE	SA
Action:		
Expectation:	*Motor x runaway = TRUE*	SF7.SF3_MonitorParameters.SF734_MonitorDroneMotors.Motro1Runaway = TRUE
Result:		SF7.SF3_MonitorParameters.SF734_MonitorDroneMotors.Motro1Runaway = TRUE 

Status:		
Note (opt.):		

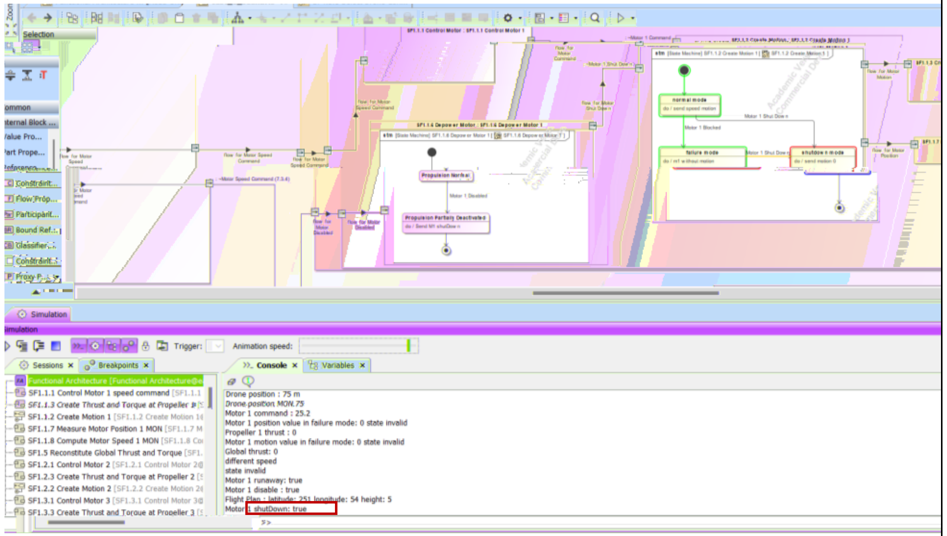
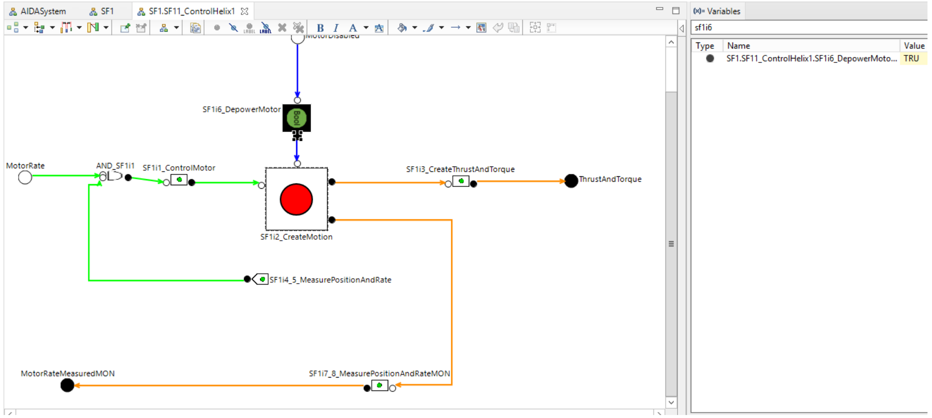
(vii) AIDA-XX.YYY-SA-007

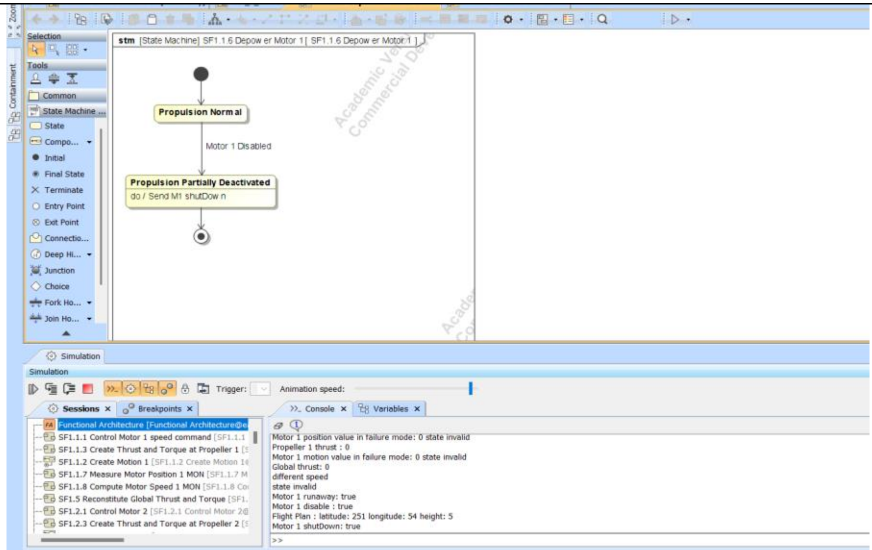
	SE	**SA**
Action:		
Expectation:	*Motor x disabled = TRUE*	**SF7.SF73.MonitorParameters.MotorsDisabled_MotorX = TRUE**
Result:		**SF7.SF73.MonitorParameters.MotorsDisabled_MotorX = TRUE** 

		
Status:		
Note (opt.):		

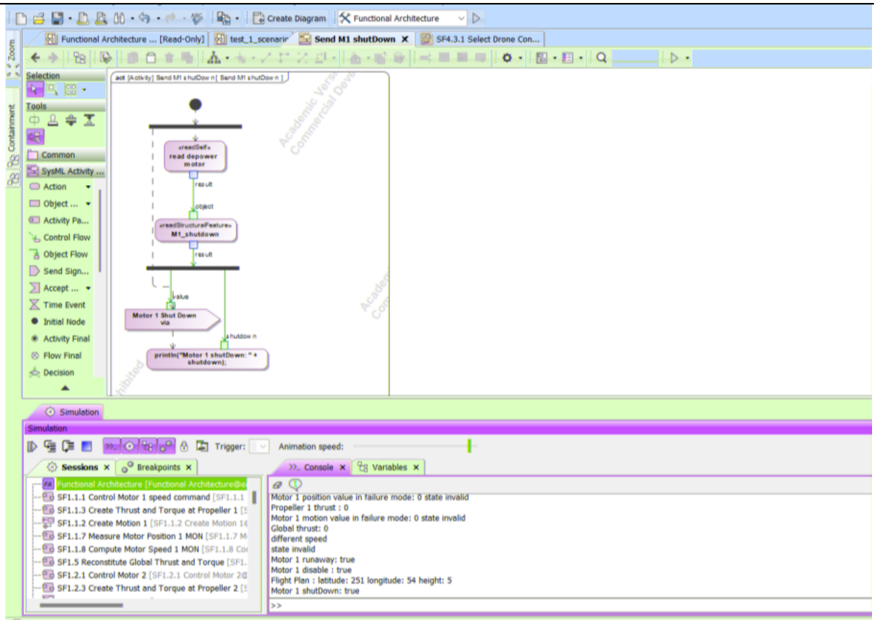
(viii) AIDA-XX.YYY-SA-008

	SE	SA
Action:		
Expectation:	Motor x Shutdown = TRUE	SF1.SF11_ControlHelix1.SF1i6_DepowerMotor = TRUE

Result:		**SF1.SF11_ControlHelix1.SF1i6_DepowerMotor = TRUE** 

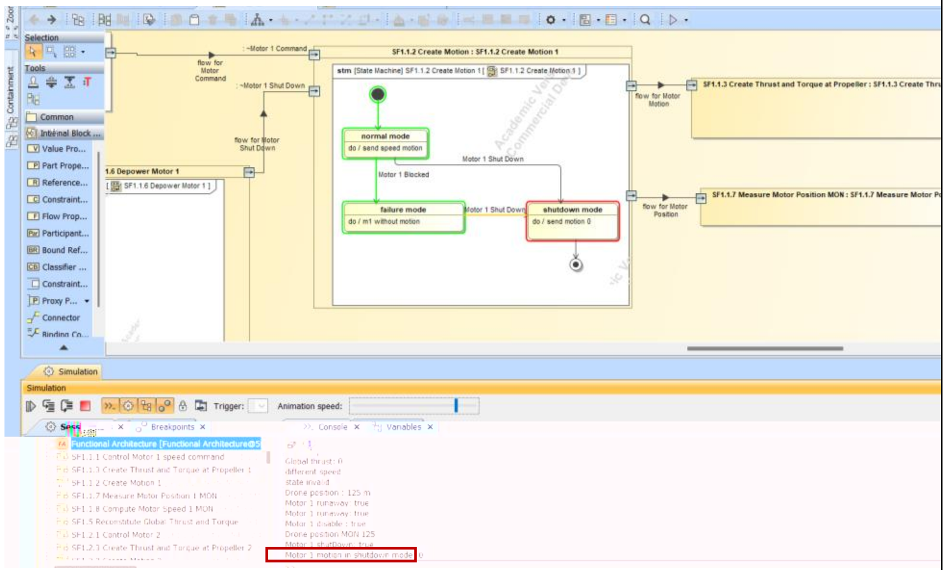
		

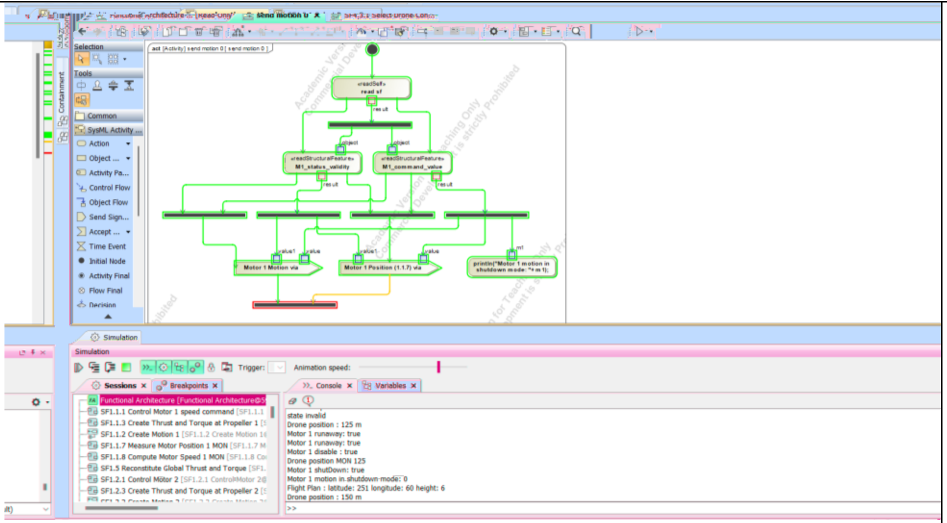
stm [State Machine] SF1.1.6 Depow er Motor 1[SF1.1.6 Depow er Motor 1]
Propulsion Normal
Motor 1 Disabled
Propulsion Partially Deactivated
do / Send M1 shutDow n
Selection
Tools
Common
State Machine ...
State
Initial
Final State
Terminate
Entry Point
Exit Point
Junction
Choice
Simulation
Trigger:
Animation speed:
Sessions
Breakpoints
Console
Variables
Functional Architecture [Functional Architecture@
SF1.1.1 Control Motor 1 speed command
SF1.1.3 Create Thrust and Torque at Propeller 1
SF1.1.2 Create Motion 1
SF1.1.7 Measure Motor Position 1 MON
SF1.1.8 Compute Motor Speed 1 MON
SF1.5 Reconstitute Global Thrust and Torque
SF1.2.1 Control Motor 2
SF1.2.3 Create Thrust and Torque at Propeller 2
Motor 1 position value in failure mode: 0 state invalid
Propeller 1 thrust : 0
Motor 1 motion value in failure mode: 0 state invalid
Global thrust: 0
different speed
state invalid
Motor 1 runaway: true
Motor 1 disable : true
Flight Plan : latitude: 251 longitude: 54 height: 5
Motor 1 shutDown: true

<table>
<tr><td></td><td>
</td><td></td></tr>
<tr><td>Status:</td><td></td><td></td></tr>
<tr><td>Note (opt.):</td><td></td><td></td></tr>
</table>

(ix) AIDA-XX.YYY-SA-009

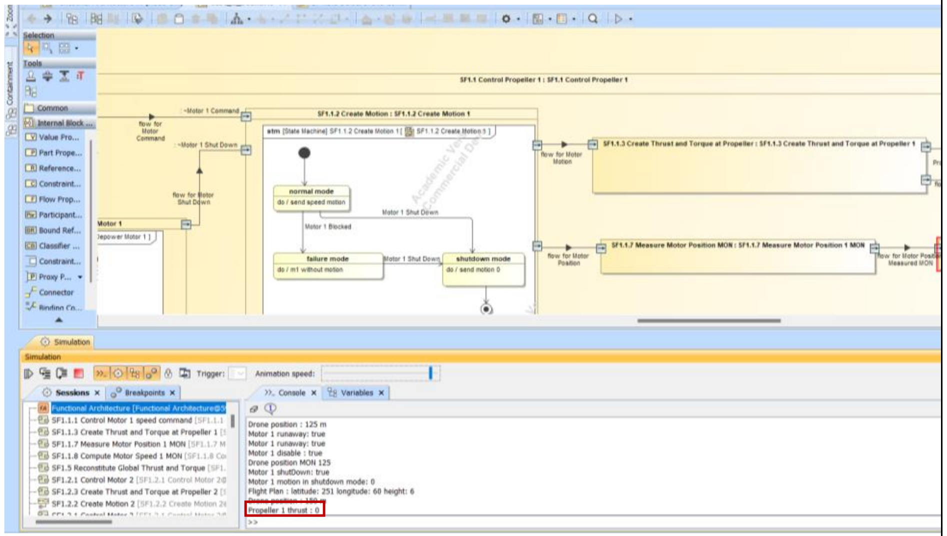
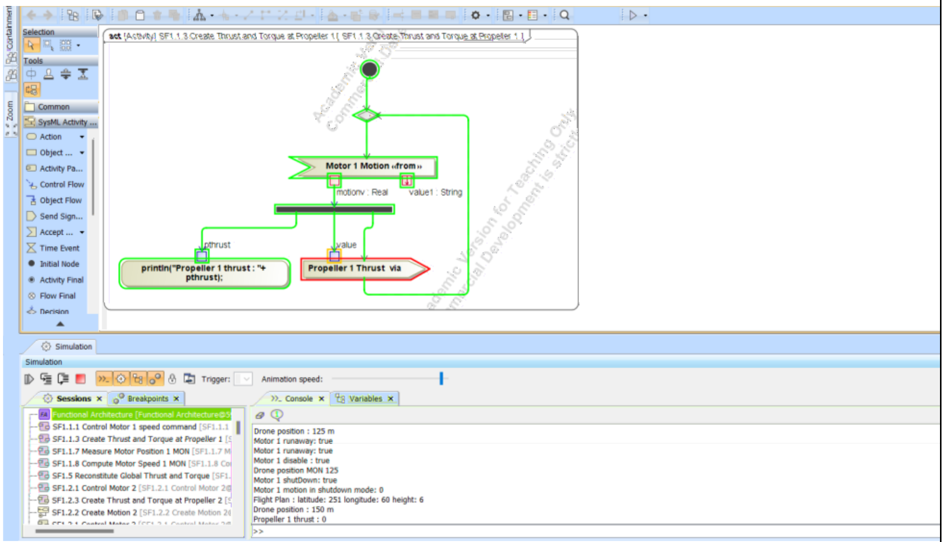
	SE	**SA**
Action:		
Expectation:	*Motor x motion = 0*	

Result:		

		
Status:		
Note (opt.):		

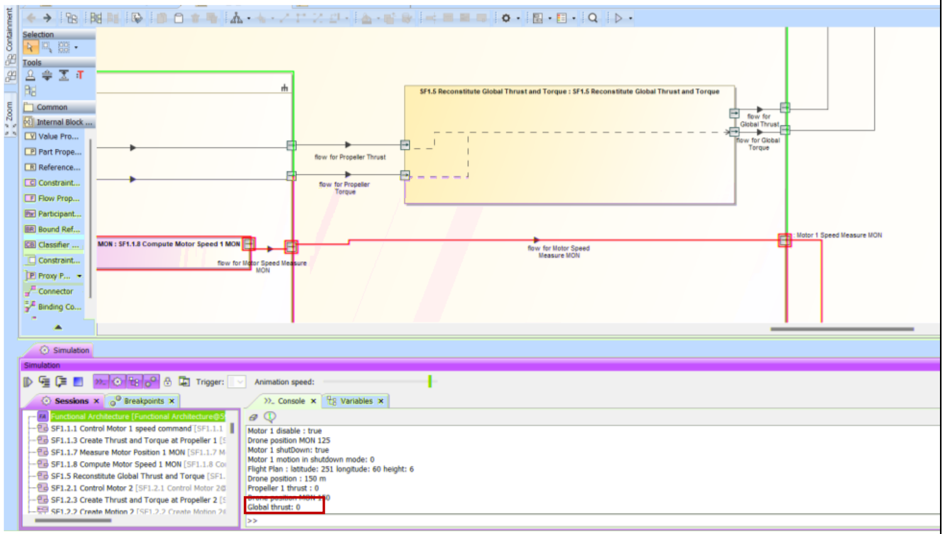
(x) AIDA-XX.YYY-SA-010

	SE	SA
Action:		
Expectation:	*Propeller x thrust = 0*	

Result:	 	

Status:		
Note (opt.):		

(xi) AIDA-XX.YYY-SA-011

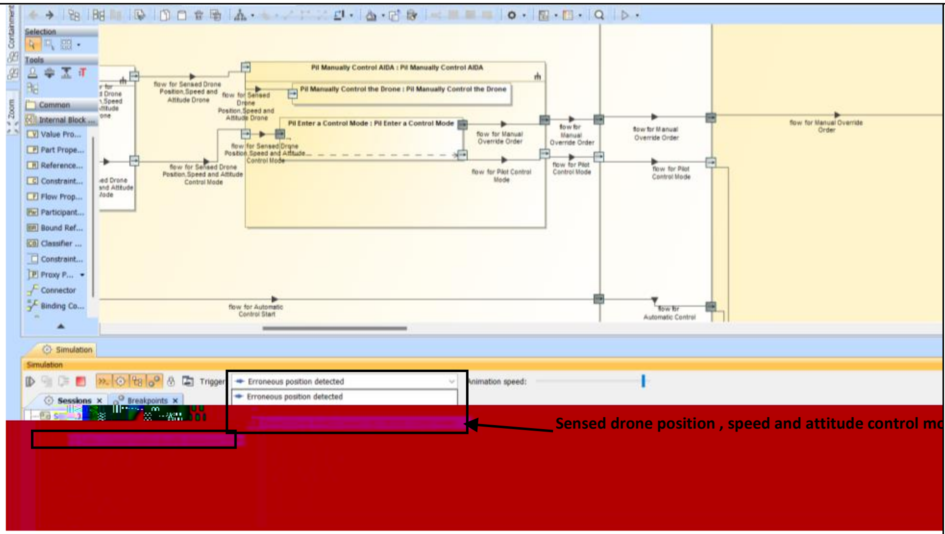
	SE	SA
Action:		
Expectation:	Global thrust! = Required Thrust +/- acceptance margin	
Result:		
Status:		
Note (opt.):		

(xii) AIDA-XX.YYY-SA-012

	SE	SA
Action:		
Expectation:	**Drone real position = (x,y,z position vector value)**	
Result:		
Status:		
Note (opt.):		

(xiii) AIDA-XX-YYY-SA-013

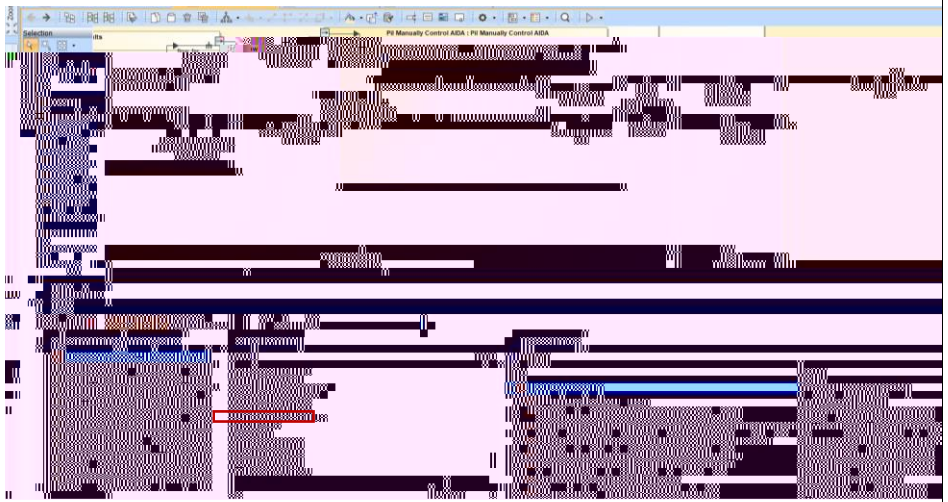
	SE	SA
Action:		
Expectation:	**Sensed drone position= (x,y,z position vector value)**	
Result:		

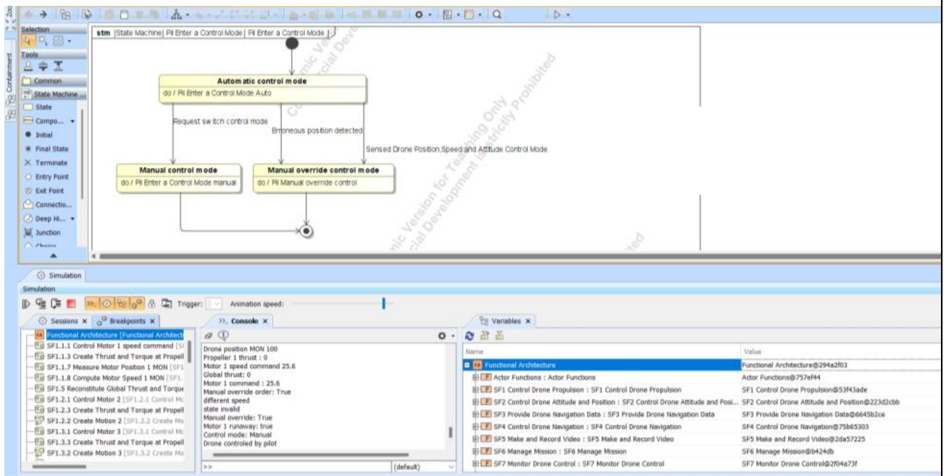
		
Status:		
Note (opt.):		

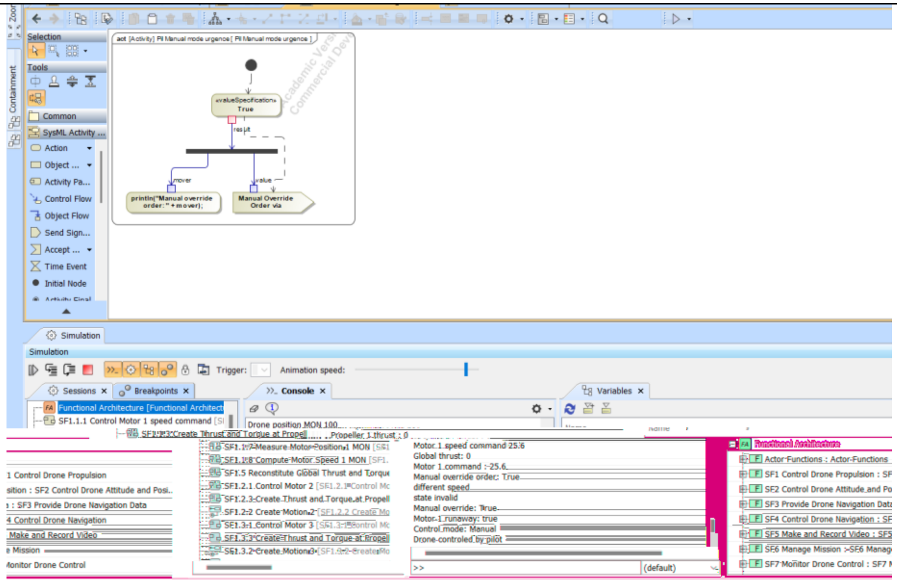
(xiv) AIDA-XX.YYY-SA-014

	SE	SA
Action:		
Expectation:	**Manual Override Order = TRUE**	

Result:

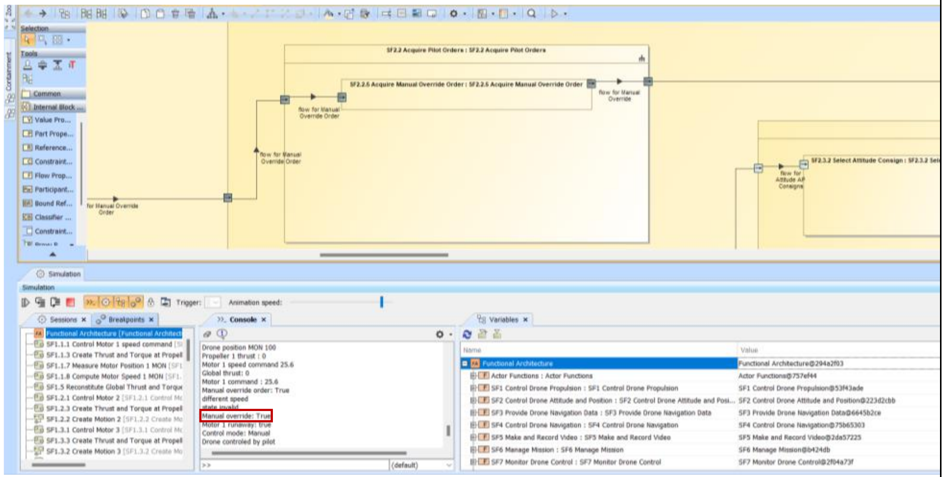
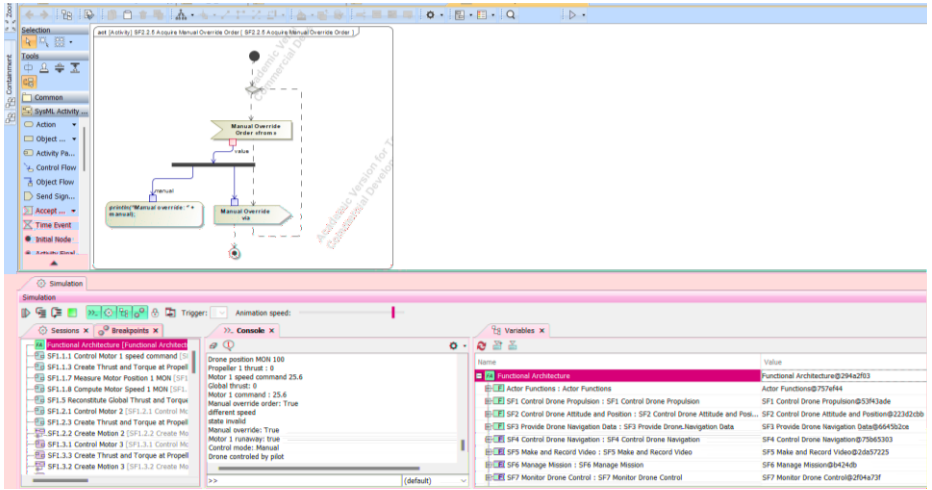



Automatic control mode
do / PI Enter a Control Mode Auto
Request switch control mode
Erroneous position detected
Sensed Drone Position Speed and Attitude Control Mode
Manual control mode
do / PI Enter a Control Mode manual
Manual override control mode
do / PI Manual override control

		
Status:		
Note (opt.):		

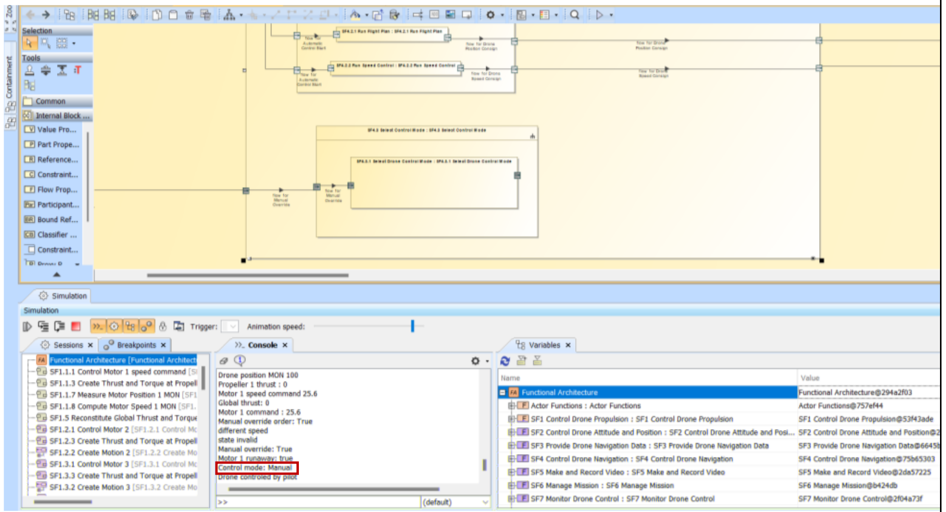
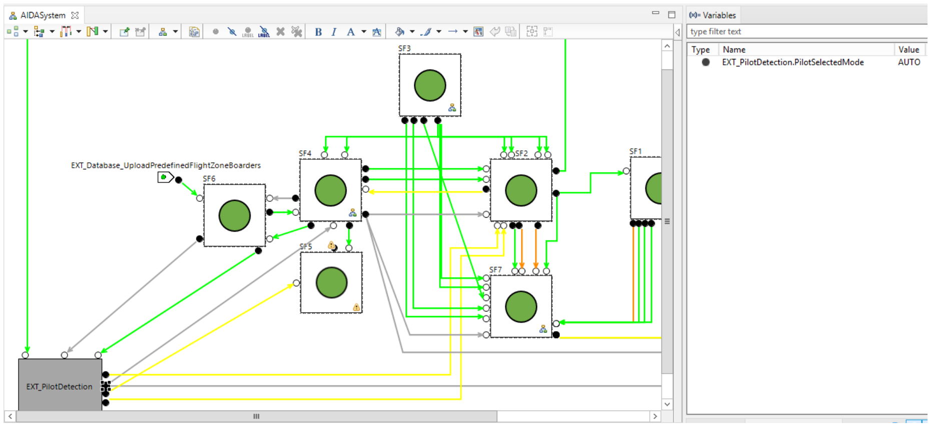
(xv) AIDA-XX.YYY-SA-015

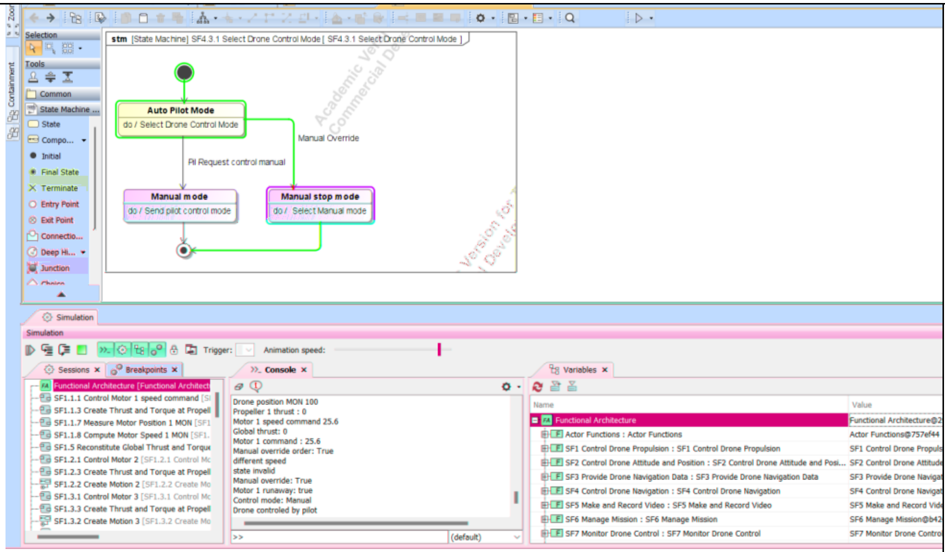
	SE	SA
Action:		
Expectation:	Manual Override = TRUE	

Result:	 	
Status:		

Note (opt.):		

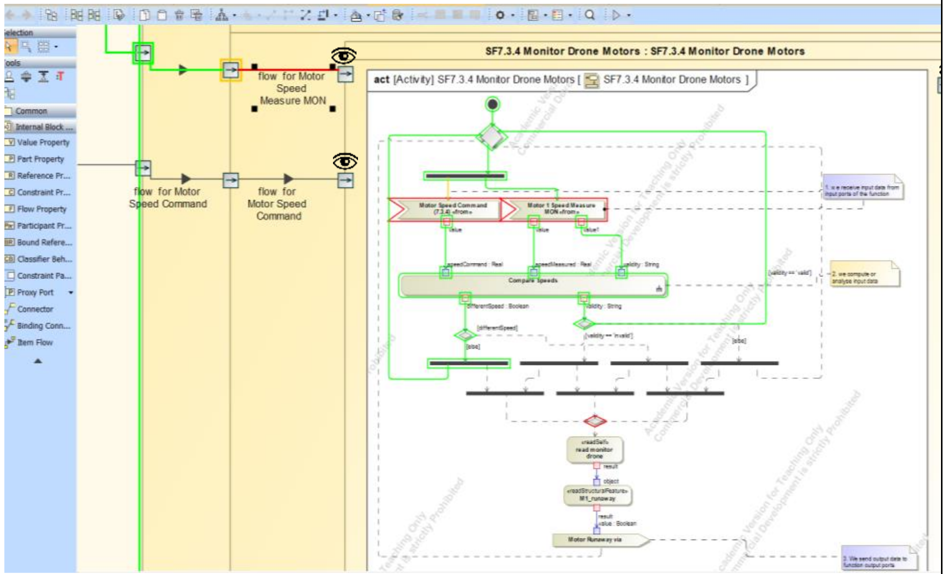
(xvi) AIDA-XX.YYY-SA-016

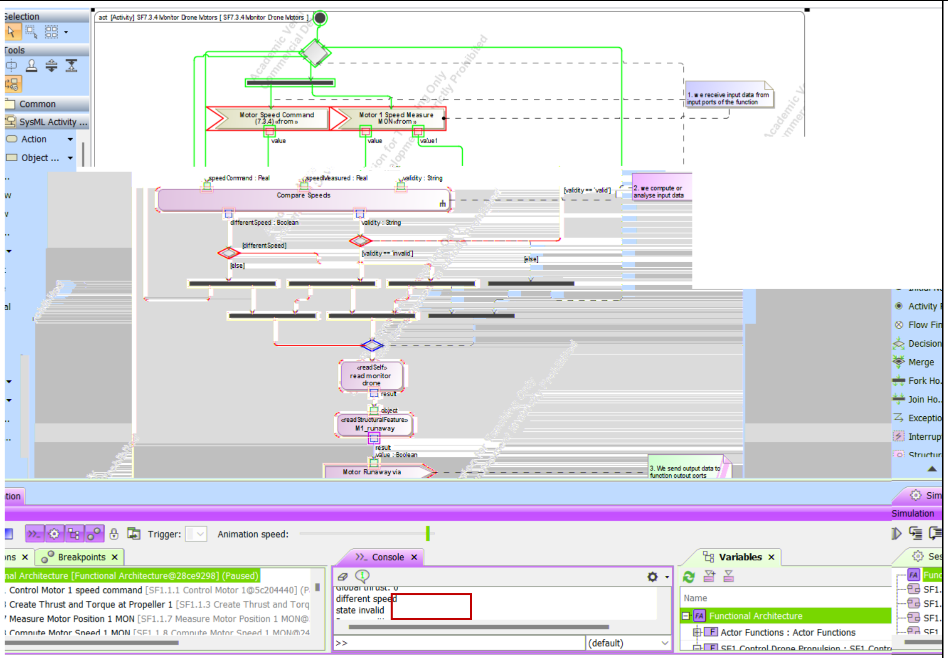
	SE	SA
Action:		
Expectation:	**Control mode = MANUAL**	**EXT_PilotDetection.PilotSelectedMode = MANUAL**
Result:		**EXT_PilotDetection.PilotSelectedMode = AUTO** 

		
Status:		
Note (opt.):		

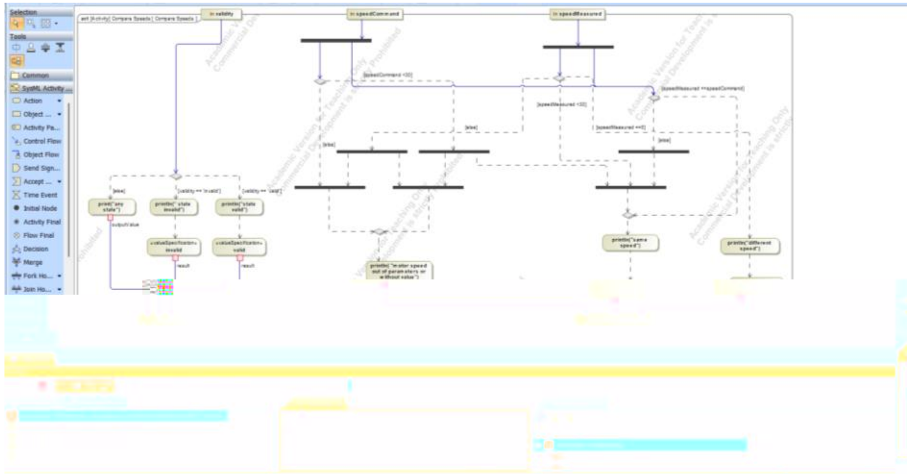
(xvii) AIDA-XX.YYY-SA-xxx (optional)

	SE	SA
Action:		

Expectation:	**Motor x speed measure COM! = Motor x speed measure MON**	
Result:		

Selection
Tools
Common
SysML Activity
Action
Object
Trigger:
Animation speed:
Breakpoints
Console
Variables
Simulation
Name
Functional Architecture
Actor Functions : Actor Functions
different speed
state invalid
(default)
Activity
Decision
Merge

		
Status:		
Note (opt.):		

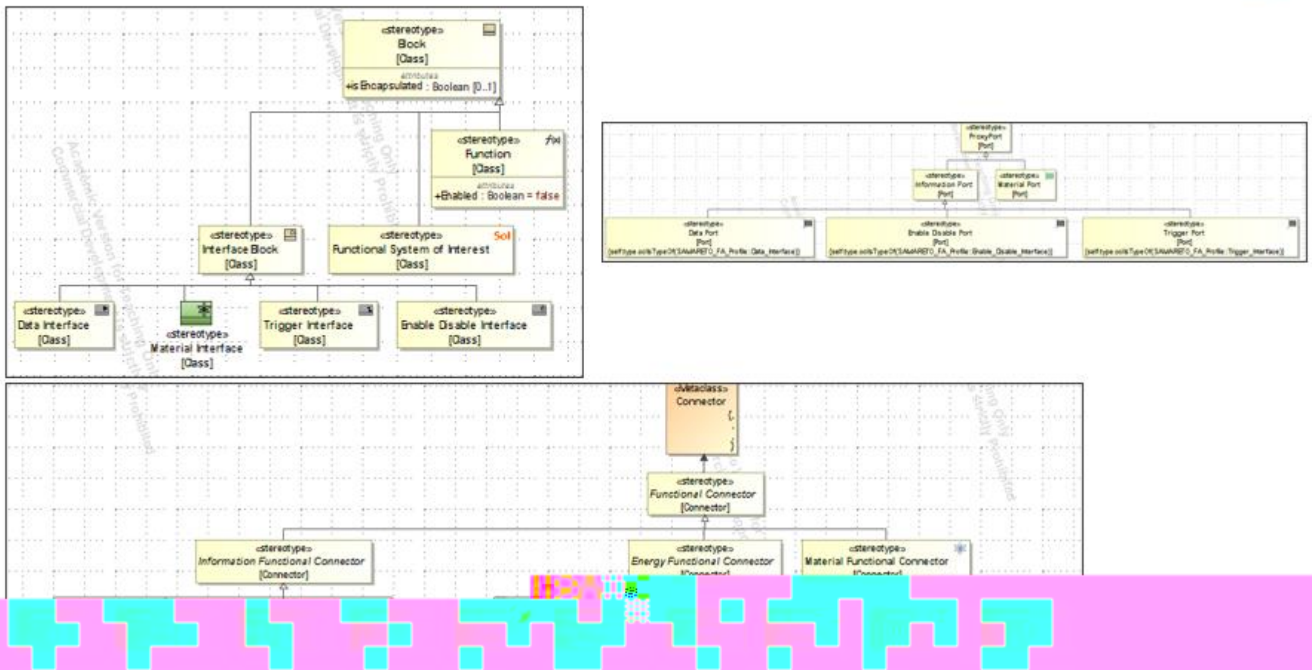
(c) PoC Conclusion

6 Conclusion

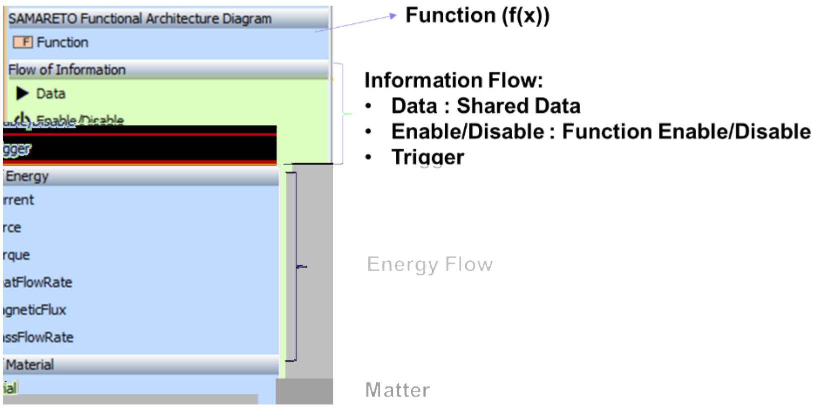
7 Appendixes

7.1 SAMAREQ Profile

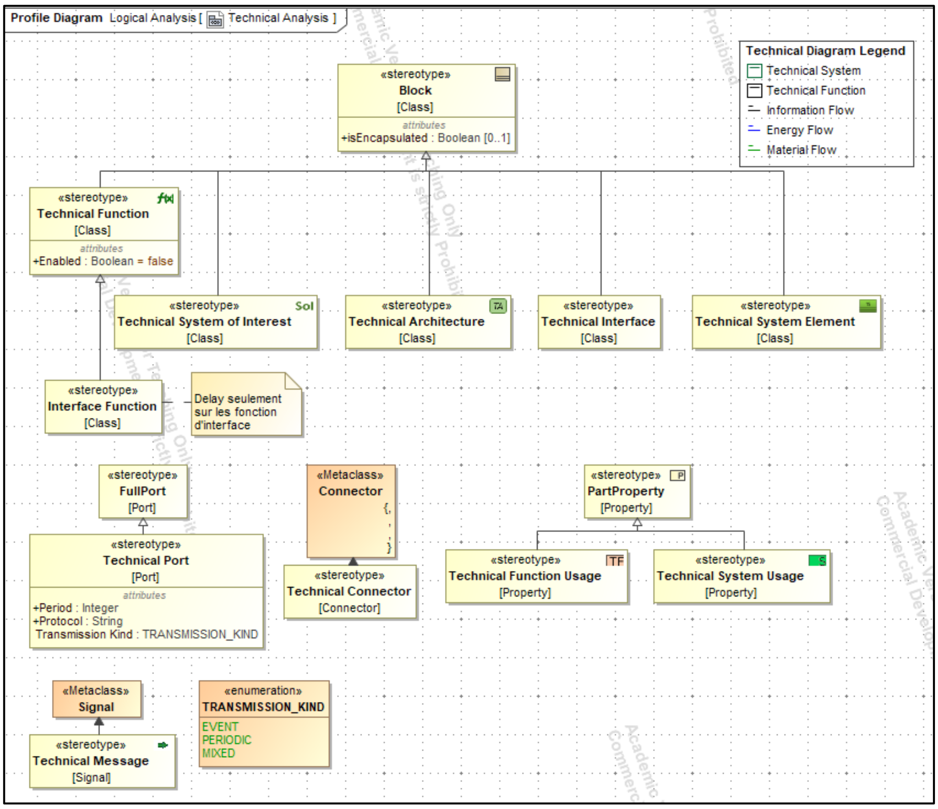
7.1.1 SAMAREQ Functional Architecture Profile



7.1.2 SAMAREQ Diagram Custoimization and Palette



7.1.3 Logical Architecture Profile



7.2 Derived Result : Consistency Check Capella Model and Cameo SysML Model