ABZ 2024

Modelling a Mechanical Lung Ventilation System using TASTD

Alex Rodrigue Ndouna, Marc Frappier

Plan

Plan

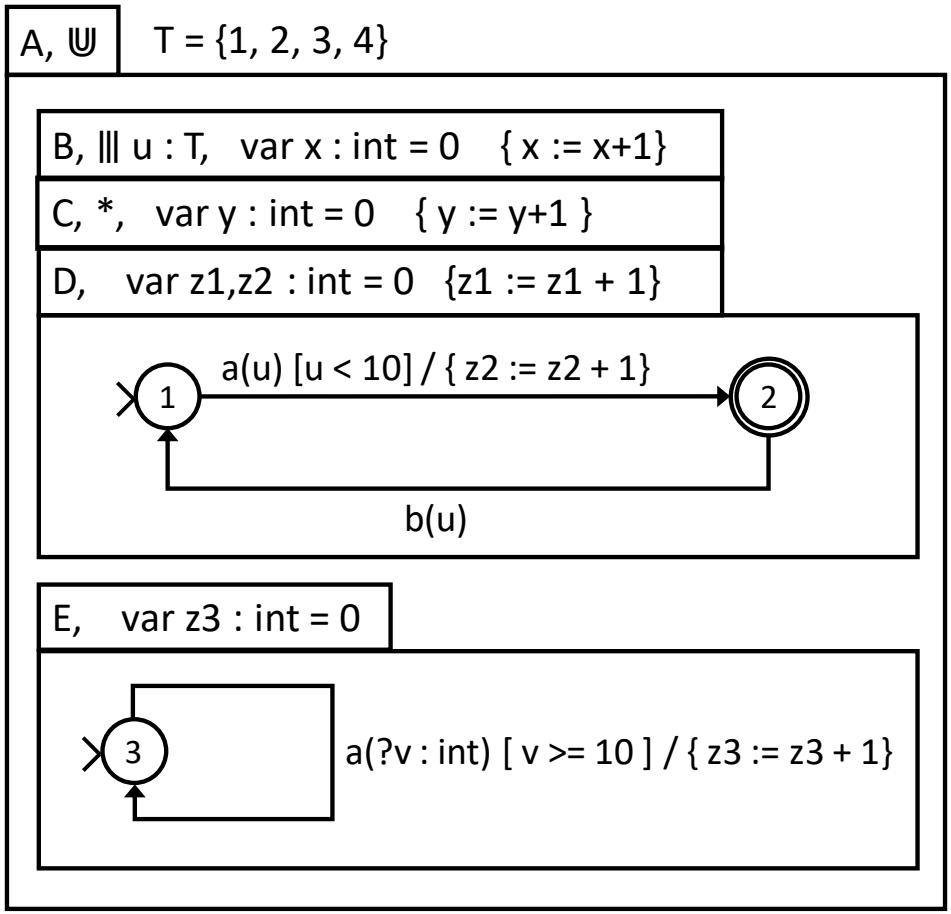
An Overview of TASTD

- Combination of state-transition diagrams *à la* state-charts, with process algebra operators *à la* CSP
- Graphical representation, Hierarchy, Orthogonality, Compositionality, Abstraction, Time
  - Transition triggered by an external event or a clock tick (Step)
- Tools : compiler cASTD, graphical editor eASTD
  - in development : invariant PO generation, Event-B Theory of ASTD
  - deprecated tools : ASTD2B, iASTD
- Case studies : ABZ Landing Gear, Mercedes, MLV, Cybersecurity (anomaly/intrusion detection)

TASTD Operators

Automaton	Sequence
Choice	Kleene Closure
Synchronization	Flow
Quantified choice and synchronization	
(Persistent) Guard	Interrupt
(Persistent) Delay	
(Persistent) Timeout	
Timed Interrupt	Call

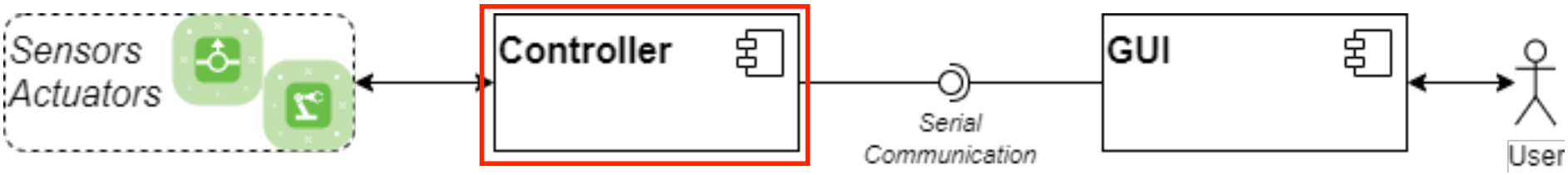
A Simple Example



			u=1			u=2			
	Event	x	y	$z1$	$z2$	y	$z1$	$z2$	$z3$
0	Init.	0	0	0	0	0	0	0	0
1	*a(1)*	1	1	1	1				
2	*b(2)*								
3	*b(1)*	2	2	2					
4	*a(1)*	3	3	3	2				
5	*a(2)*	4				1	1	1	
6	*a(1)*	5	4	1	1				
7	*b(1)*	6	5	2					
8	*b(2)*	7				2	2		
9	*a(10)*								1

Plan

What was Modeled



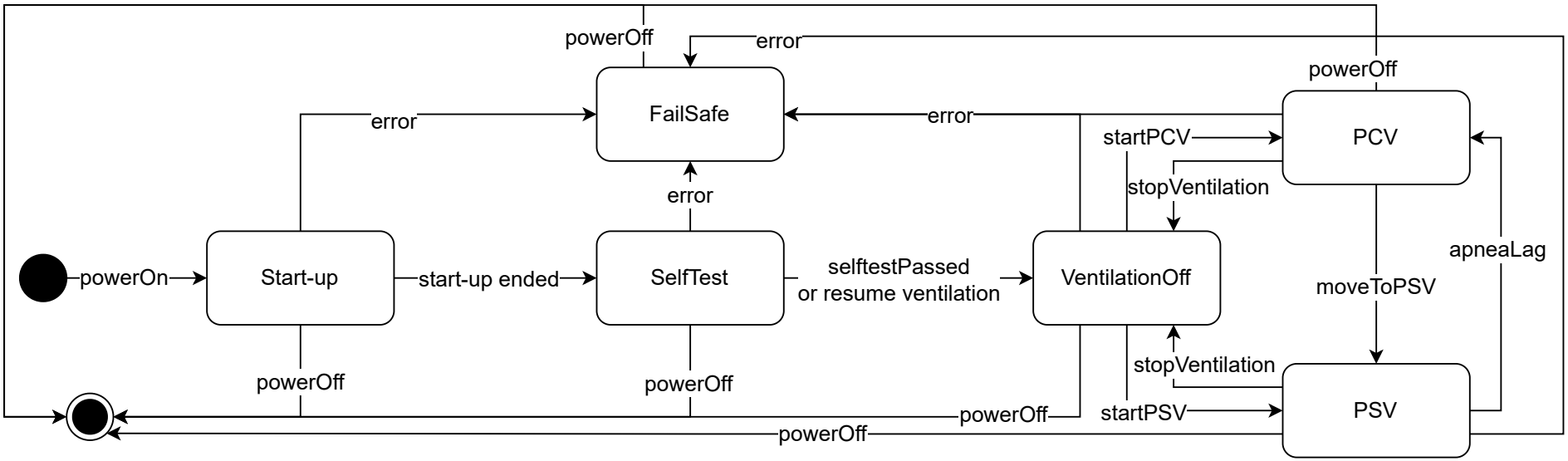


Figure 4.1: Controller state machine

How was it Modeled

- Decompose the system into small components
  - Complex states of the case study state machine
  - Sensors and actuators
- Specify and validate each component separately
- Intuition is that component composition is an alternative to the refinement à la Event-B

Sensors and Actuators

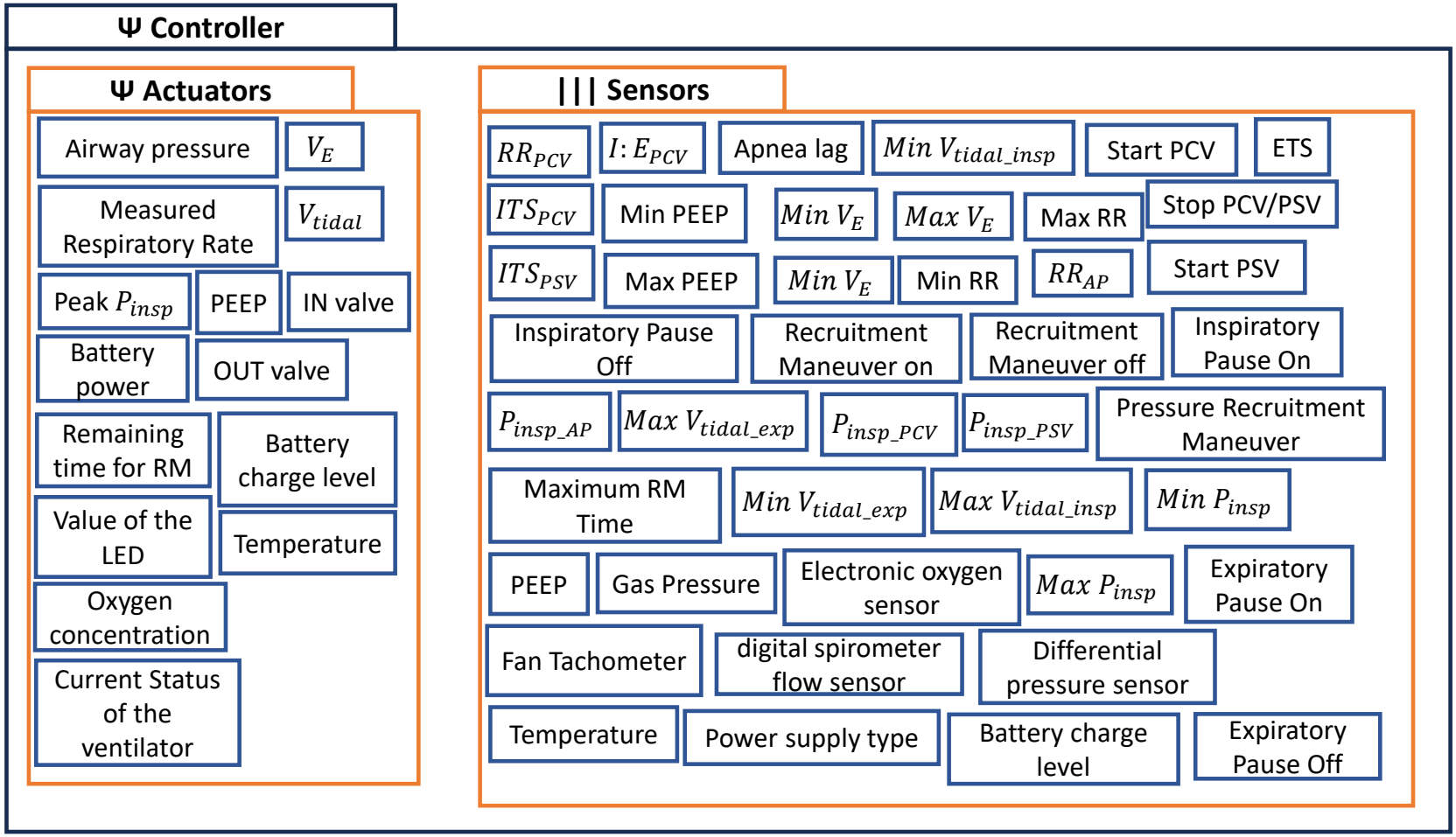


Figure: ASTD Controller composing sensors and actuators

Main ASTD

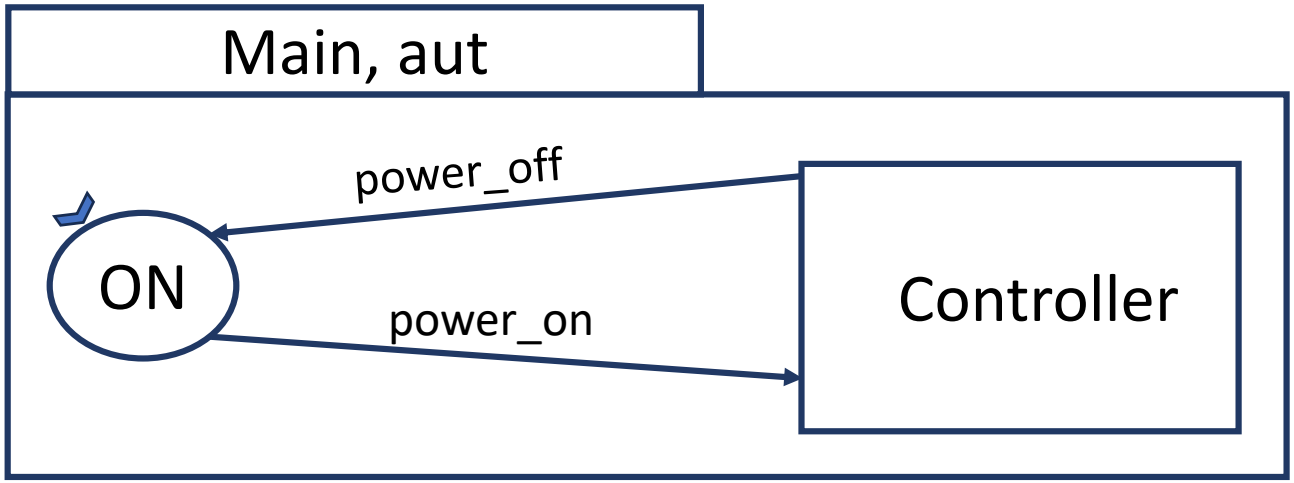


Figure: Main ASTD of Mechanical Lung ventilator controller

Controller ASTD

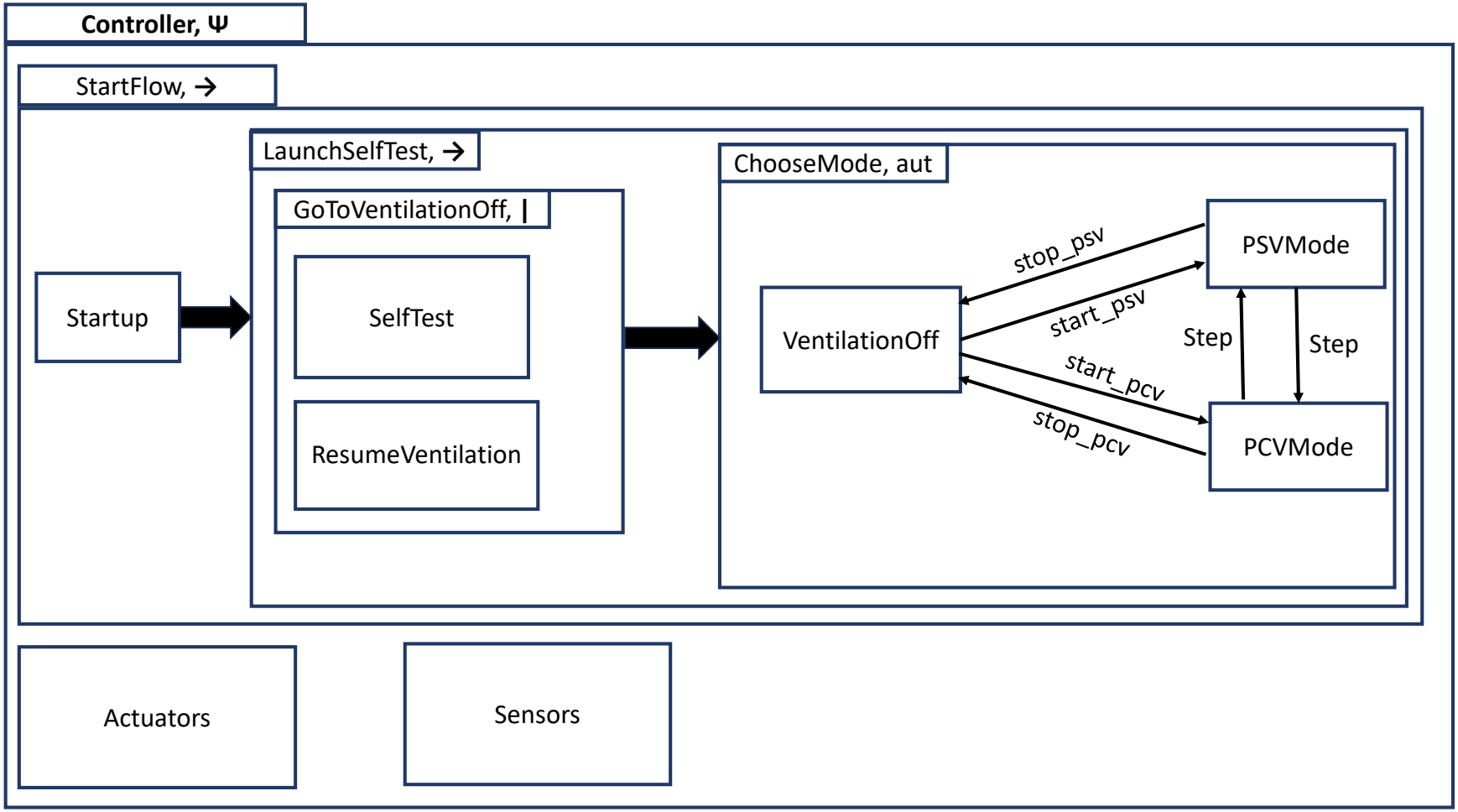


Figure: Controller ASTD of Mechanical Lung ventilator

Communication with shared variables

Component	Variables
Main (root)	*parameters*, *power_status*
Controller	*error_value*, *paw_drop_in*, *last_inspiration_time*, *move_to_pcv*, *mode*, *peak_pinsp*, *ve*, *rr*, *peep*, *fio2*, *vtidal*, *updated_parameters*
StartUp	*adc_timeout*, *pressure_sensor_timeout*, *error*, *processes_count*
InspirationPhaseEnd	*inspiration_phase_timer*
Initialization	*attempt*, *status*
CheckSensors	*pressure_sensor_valid_response*
PCVModeExpiration	*its_trigger_window*
PSVModeExpiration	*its_trigger_window*, *inspiration_time*
LaunchLoop	*enable_loop*

Table: Shared variable by components

Formalization of the requirements

ASTD	Requirements
Controller	CONT.1, CONT.2, CONT.3, CONT.4, CONT.5, CONT.6, CONT.7, CONT.8, CONT.9, CONT.10
StartUp	CONT.12, CONT.13, CONT.14, CONT.15, CONT.16
SelfTest	CONT.17, CONT.18, CONT.19
VentilationOff	CONT.38
PCVMode	CONT.20, CONT.21, CONT.22, CONT.23, CONT.24, CONT.25, CONT.26, CONT.27, CONT.28, CONT.39, CONT.40, CONT.41, CONT.42, CONT.43, CONT.44, CONT.45
PSVMode	CONT.29, CONT.30, CONT.31, CONT.32, CONT.33, CONT.34, CONT.35, CONT.36, CONT.37, CONT.39, CONT.40, CONT.41, CONT.42, CONT.43, CONT.44, CONT.45
FailSafe	CONT.38
Main (root)	CONT.11, CONT.38

Table: Cross-reference between ASTDs and requirements for mechanical lung

Statistics

- Modeling time : 80 h
- Validation time : 40 h
- 160 ASTDs

73 automaton	28 call
5 sequence	15 closure
13 guard	7 choice
5 synchronisation or interleaving	4 flow
2 delay	8 timeout

Plan

Sensors ASTD

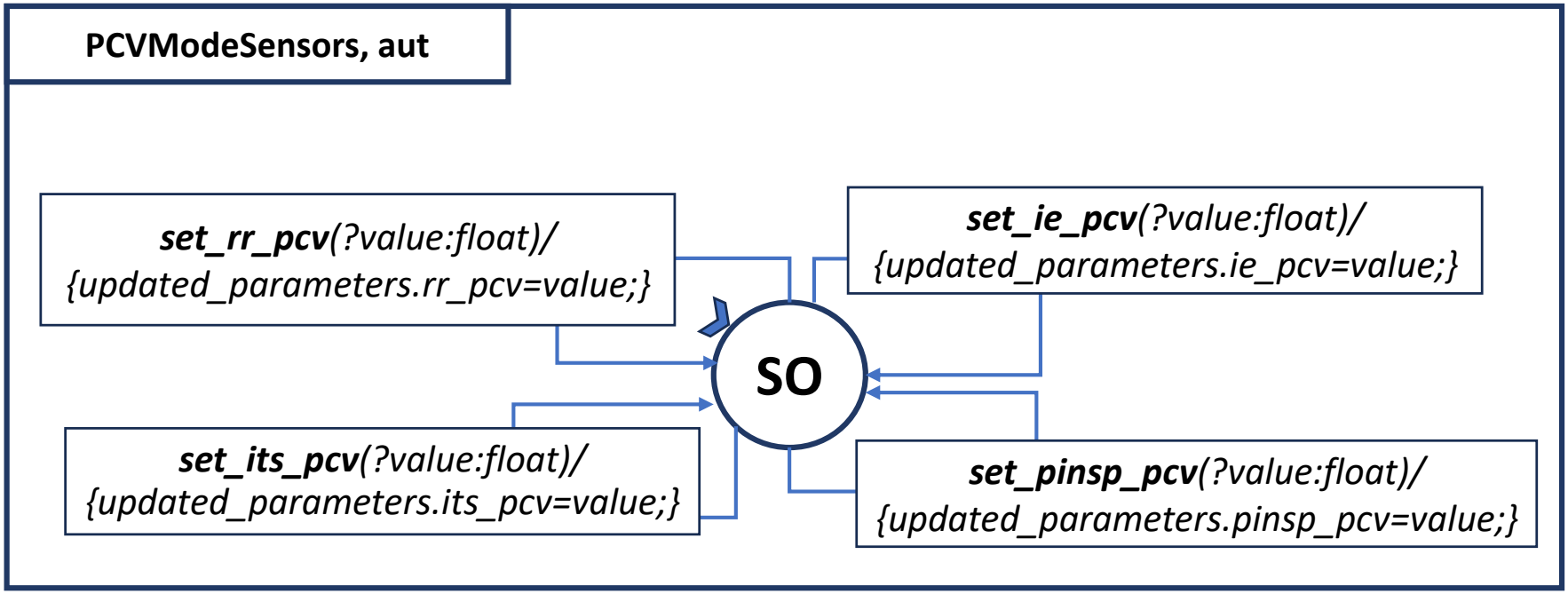


Figure: Automaton ASTD PCVModeSensors

Actuators ASTD

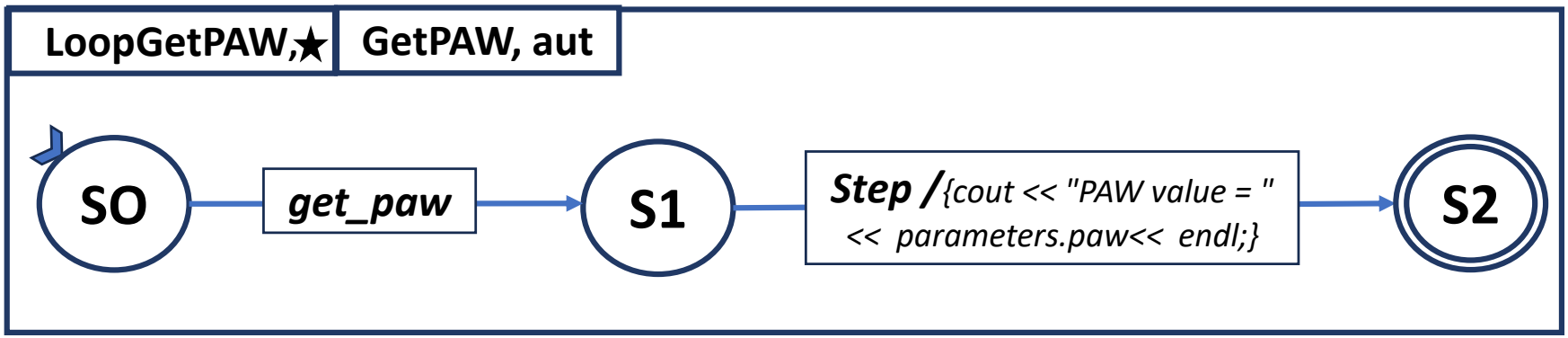


Figure: Automaton ASTD LoopGetPAW

Main ASTD

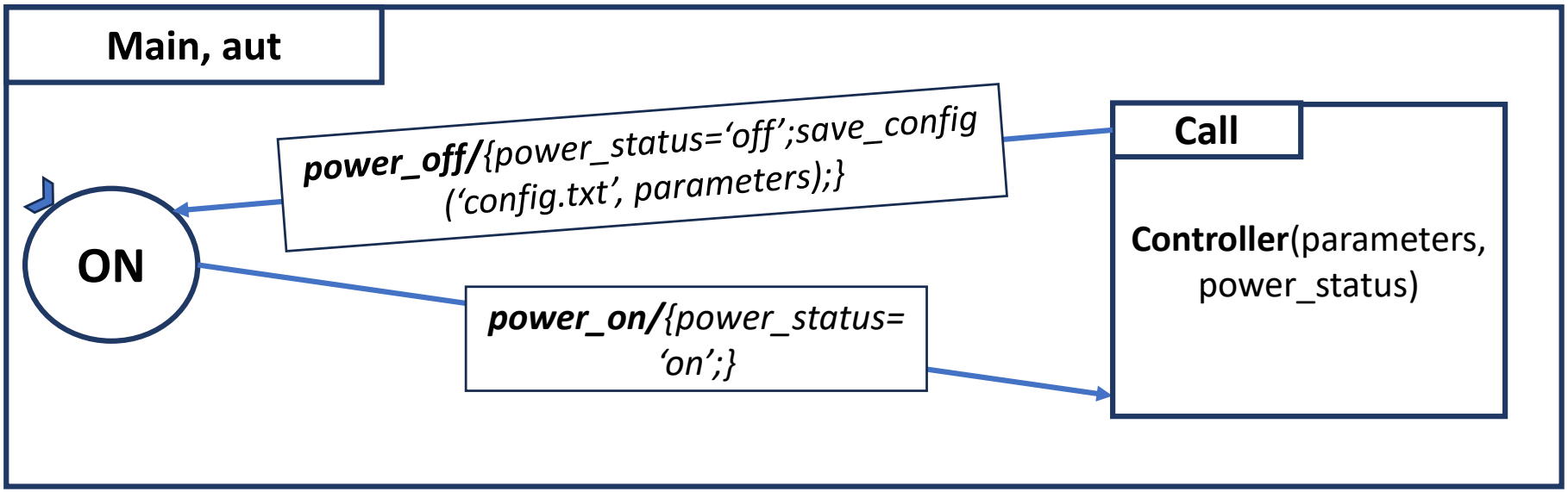


Figure: Automaton ASTD Main

Controller ASTD

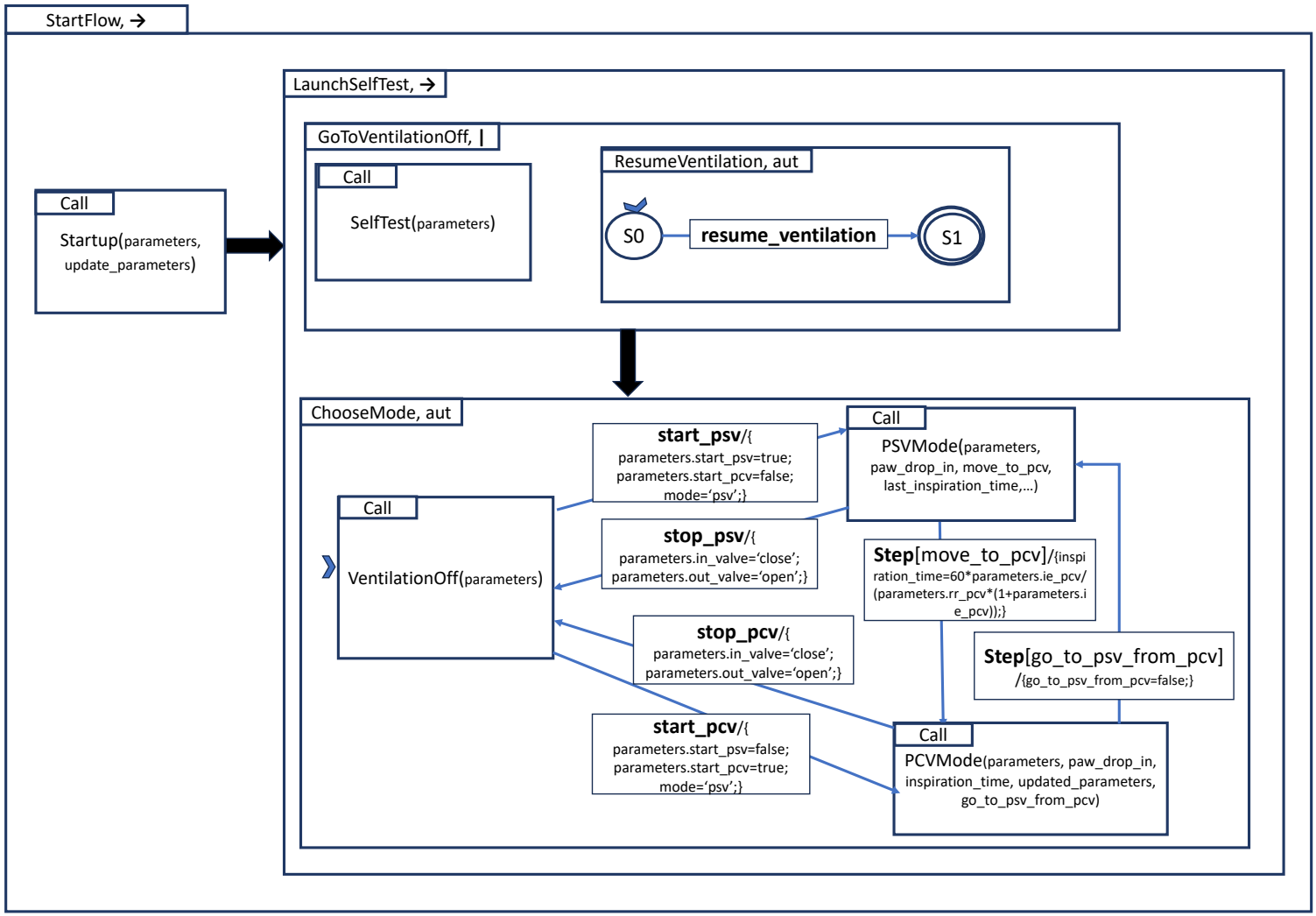


Figure: StartFlow ASTD

Modelling Time requirements(CONT.22)

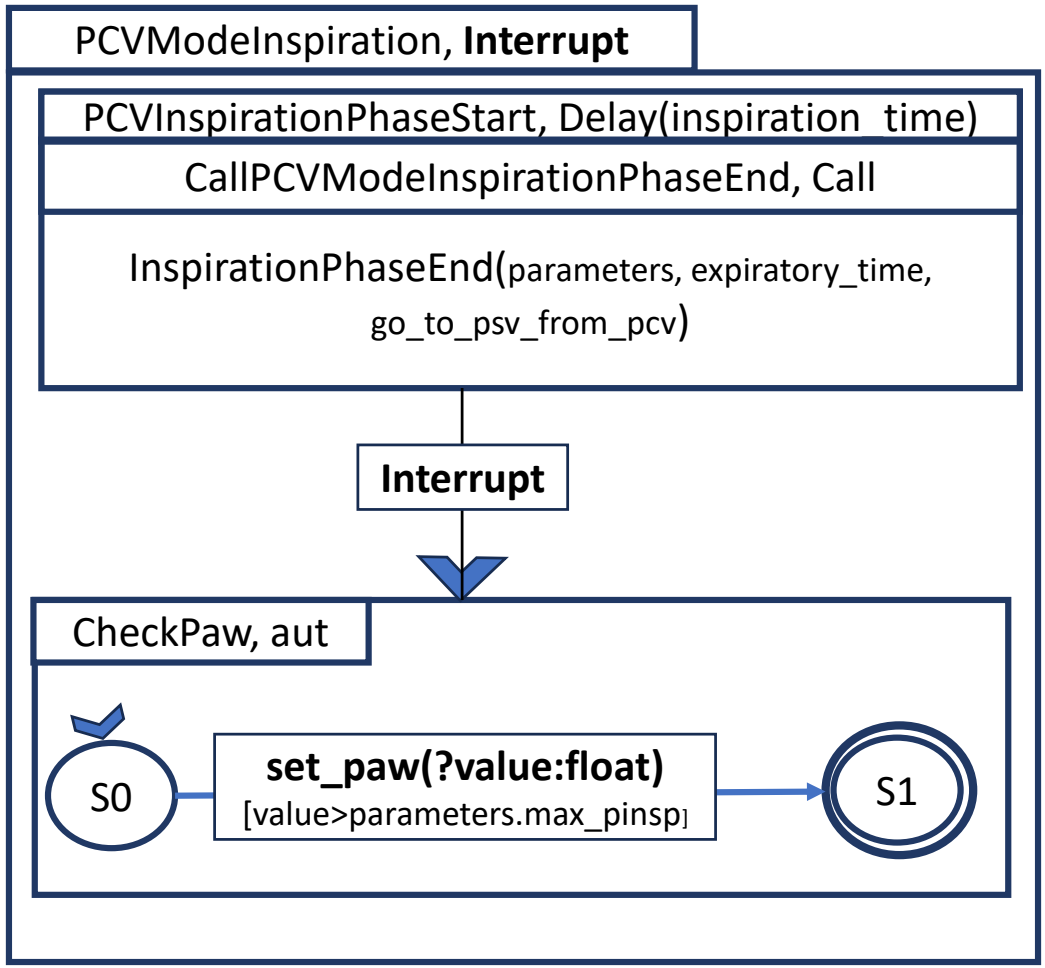


Figure: PCVModeInspiration ASTD

Modelling Time requirements(CONT.42.2)

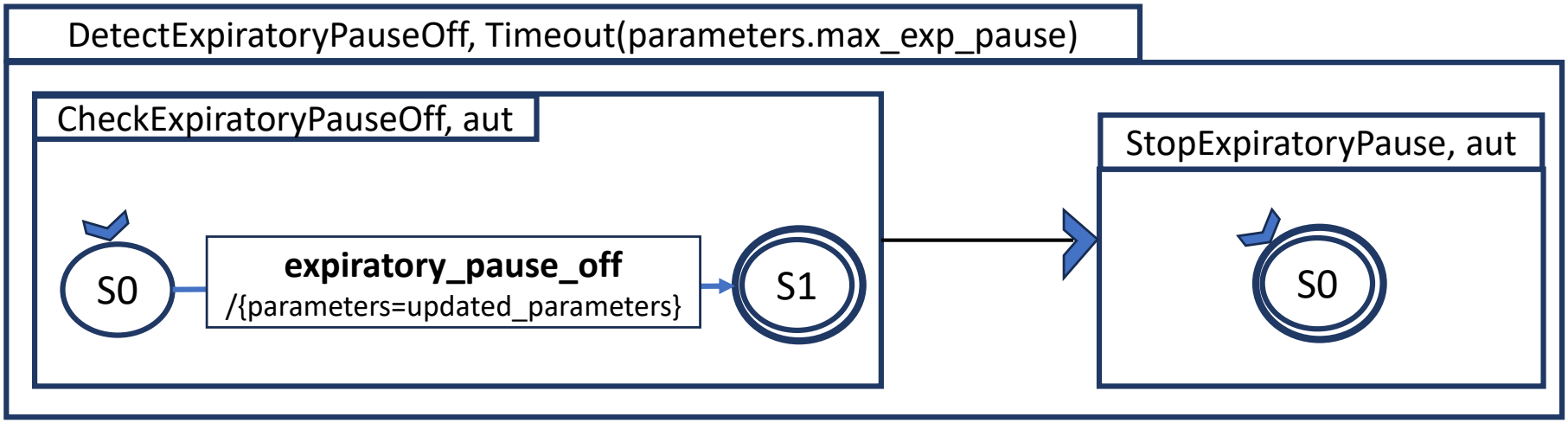


Figure: DetectExpiratoryPauseOff ASTD

Plan

Validation and Verification

- use interactive animation of the specification with the executable code generated by the cASTD compiler
- compilation is automatic, and no human modification is necessary on generated code
- We have implemented 12 test cases
  - 5 PSV mode scenarios
  - 6 PCV mode scenarios
  - 1 start-up scenario up to VentilationOff step

Validation Issues

- Large number of ASTDs
- Compiler cASTD fails to generate code for the whole specification (Java memory overflow)
- Generated code larger than the Mercedes Case Study of ABZ2020
- Inefficient code generation (lack of modularity, code duplication, long prefix names to access nested ASTD attributes, etc)

Plan

Specification Ambiguities and Flaws

- Very few (small) issues
- Controller requirements : ambiguity or lack of information
- SelfTest step
  - FUN.6, CONT.17, CONT.18, CONT.19
  - no indication on how the controller receives self-tests results
- Verification of sensor communication
  - CONT.15 states that a maximum of 5 connection attempts must be made with the pressure sensor
  - after 5 attempts, assume that that there is an error and the controller switches to FailSafe mode
  - no indication of how long to wait before attempting a new connection.
  - Same issue with CONT.16 - initialize external ADC
- We defined them as system parameters

Plan

Conclusion

- Complete model of the controller component
- Reuse specification style of Mercedes case study ABZ2020
- Extensive use of modularity / decomposition
- Validation by animation using the compiler generated implementation of the specification
- Compiler needs to be optimized to handle larger (modular) specification

Future Work

- Optimise the compiler
- Develop verification techniques
  - invariant proof obligations (done for 4 operators)
  - temporal constraints (CCSL, reachability, leads to)
  - use Event-B ASTD metamodel
- compare ASTD spec with Event-B spec of Amel Mammar
  - evaluate impact of modularity on proving
  - compare how properties can be specified and verified

Questions