Introducing Feasible Safety to Autonomous Firefighting Drone

Workshop: Exploring Formal Methods for Unmanned Aerial Vehicles

Paulo Bezerra, Ana Cavalcanti, Thierry Lecomte , Pedro Ribeiro

ROBOSTAR

robostar.cs.york.ac.uk

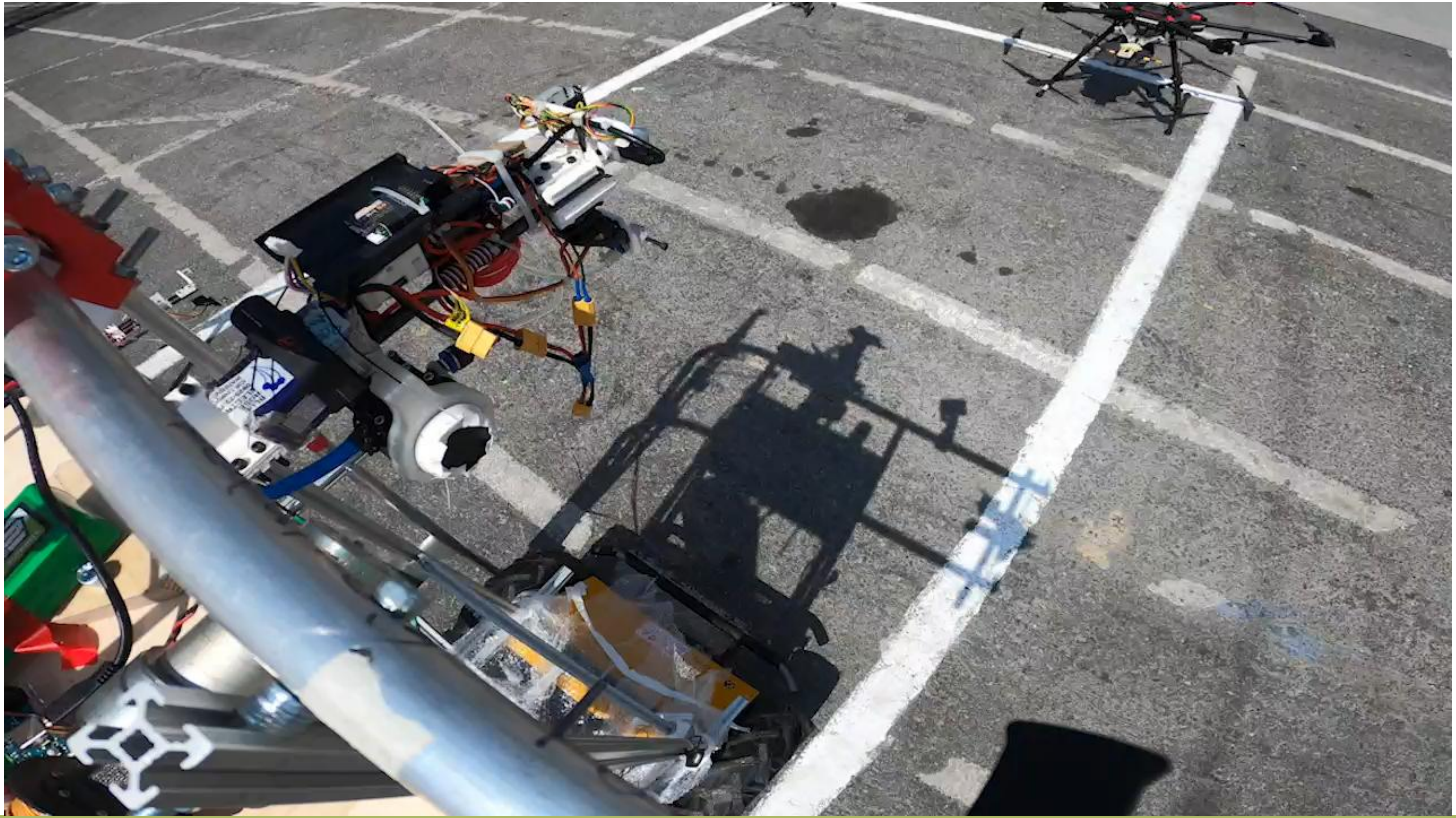
10th June 2025

Royal Academy of Engineering

Engineering and Physical Sciences Research Council

THE SUBJECT

How to add safety to this firefighter drone



ROBOSTAR

Failing Software-Based Systems

If it only runs on a computer screen, it is not really safety

Cybersecurity: same risks but ≠ POV

ROBOSTAR

Safety for Unmanned « flying things »

SAFETY INTEGRITY LEVELS
DESIGN ASSURANCE LEVELS

DAL B/SIL3 : 10^{-7}/h
DAL A/SIL4 : 10^{-9}/h

CATASTROPHIC FAILURES

CERTIFICATION

NL safety demonstration
Formal methods **in dev cycle**

SYSTEMATIC FAILURES

Specification
Design
Implementation
Environment
Exploitation

STRONG / ONGOING STANDARDS

General aviation standards:
DO-178C (SW), DO-254 (HW), ARP4761 (Safety)

Unmanned Aircraft Systems:
ASTM F3266-20 (Design)
ASTM F3178-16 (Loss of control)
Specific Operations Risk Assessment
European Union Aviation Safety Agency
DO-326A (Cybersecurity)

RANDOM FAILURES

Execution machine
Entropic hardware

ROBOSTAR

THE SAFETY ANALYSIS

« Safety is by design »

Hazard Analysis: Preliminary Study

- Dreaded events (what situation do we want to avoid ?)
  - **[1] Erratic flight**
    - Hypothesis: behaviour is supposed « correct »
    - Adding functional redundancy (duplicate computer, software, and sensors) against the lightweight / lowcost design principles
  - **[2] Collision with environment or human being**
    - Hypothesis: Lightweight drone -> probably no incidence
  - **[3] Loss of the drone**
    - Requires safeguard to avoid drone to get out of reach /lost

Preliminary Study

- Dreaded events (what situation do we want to avoid ?)
  - **[1] Firefighting erratic flight**
    - Hypothesis: behaviour is supposed « correct »
    - Adding functional redundancy (duplicate computer, software, and sensors) against the lightweight / lowcost design principles
  - **[2] Collision with environment or human being**
    - Hypothesis: Lightweight drone -> probably no incidence
  - **[3] Loss of the drone**
    - Requires safeguard to avoid drone to get out of reach /lost

ROBOSTAR

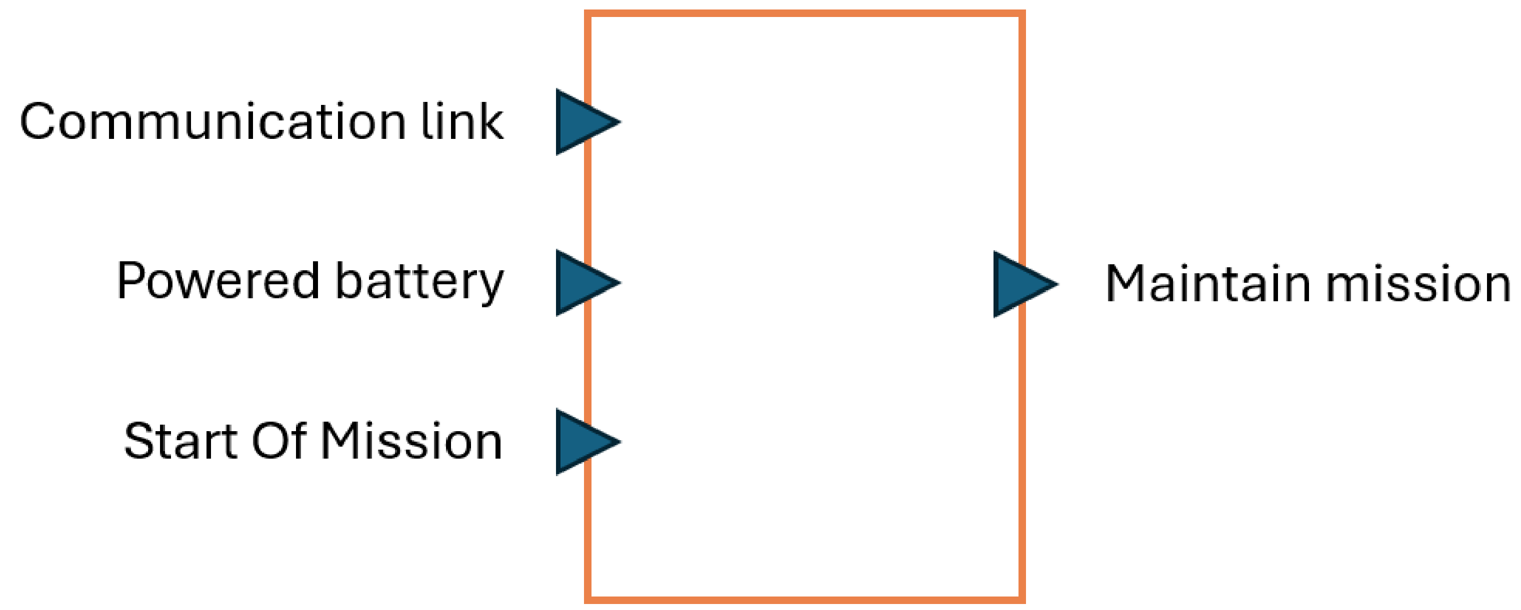
Hazard	Accidental event	Probable cause	Preventive actions
Loss of communication	Inability to control drone (mission interrupt)	ECM, fuzzing, emitter down, receiver down, obstacle, signal attenuation	If no signal is received during a given period, flight software is triggered to “return to base”
Invalid communication	Mission maintained with no valid remote control	Wrongly received signal from another source	Messages contains some **liveness** and dynamic information to discriminate from “random sources”
Low energy	Inability to maintain communication link, inability to ensure flight	Battery low, leak current	External device measures remaining charge and trigger alarm if half charge + constant is reached. Also takes into account the loss of charge over time.
Insufficient propulsion power	Inability to maintain flight profile, collision with ground objects/human beings	Environmental conditions (wind), interaction with environment (cables), engine failure	Out of scope
Inaccurate flight computer	Unpredictable trajectory, collision with objects/human beings	ECM, shots, failing hardware	Out of scope
Safety function not active	Inability to control drone (mission interrupt)	No energy on the safety computer, failing safety computer	Safe state should correspond to “safety computer powered and running OK”

ROBOSTAR

Safety Check

- **Verifying that a communication link is maintained during the whole mission**. This communication link, from ground base to drone only, is used to interrupt the mission if decided remotely by human supervisors and/or if some on-board conditions are not met. Recovering the communication link re-enables the mission.
- **Checking that the battery has sufficient charge**. Insufficient charge implies to recharge the battery of the drone that is the only way to cancel the “low battery” alarm.
- If the safety-check fails, the flight software is contained in a mode where a return-to-base is mandatory.
- **Need to know when the mission starts** (Start of Mission, or ***SoM***)
- **Operational exported constraint**: drone cannot be operated from a moving base

Safety Check



It takes three inputs:

- **Communication link** represents the transformation of analogic radio signal into digital signal (bit stream). The frequency of the signal and bit alternation is constant. The transmission pattern must be determined. When communication link is down, the mission is not maintained until either the communication link is reestablished, or the drone reaches base and is reset/restarted.

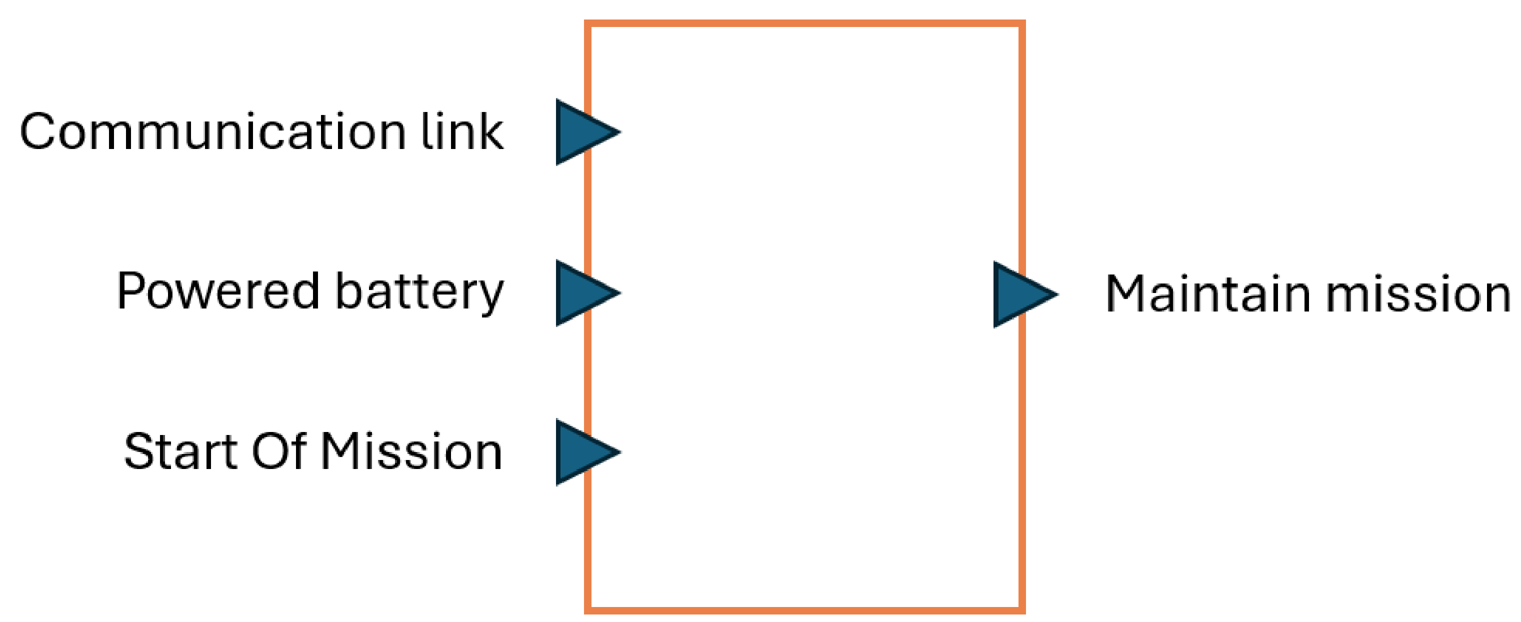
- **Powered battery** represents the capability of the drone to return to its base, as it is supposed to start its mission with full charge. The data required for the low battery alarm is usually complex (real value fluctuating over time). For this case study, the Boolean input signal represents the fact that the output voltage is greater than a threshold. If it is lower than this threshold during a delay delay$_1$ then the low battery alarm is raised. Once a low battery alarm is raised, the return-to-base is forced until the drone returns to base and is reset/reenergized/restarted

- **Start of Mission** represents the first moment when the safety check must be ensured. This event is characterized by the first rising edge of this input.

and calculates one output:

- **Maintain mission** represents the ability to continue the mission.

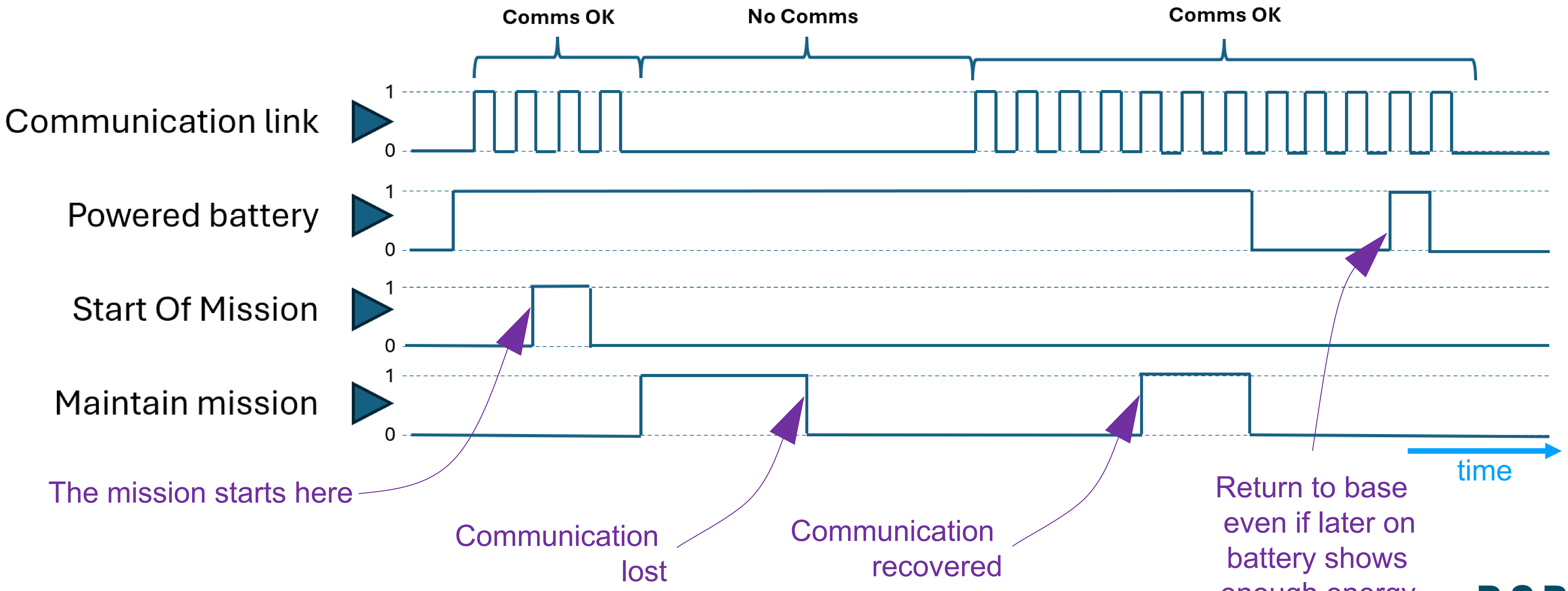
Inputs, Outputs, and Safe State



- Restrictive position (“return to base”) should correspond to “absence of power”
  - *Maintain mission* should be powered to maintain the mission
  - *Powered battery* indicates enough energy when powered
  - *Start Of Mission* requires some energy to start the mission
  - *Communication Link* not energized indicates no communication activity

ROBOSTAR

An example of scenario



THE APPROACH

Formal Methods to Handle Failing Systems

RoboSim for system level modelling

B for C&C non-threaded safety software

CLEARSY SAFETY PLATFORM

Natural language requirement

Specification

Design

Source code

Binary code

Bad hardware

Wrong specification

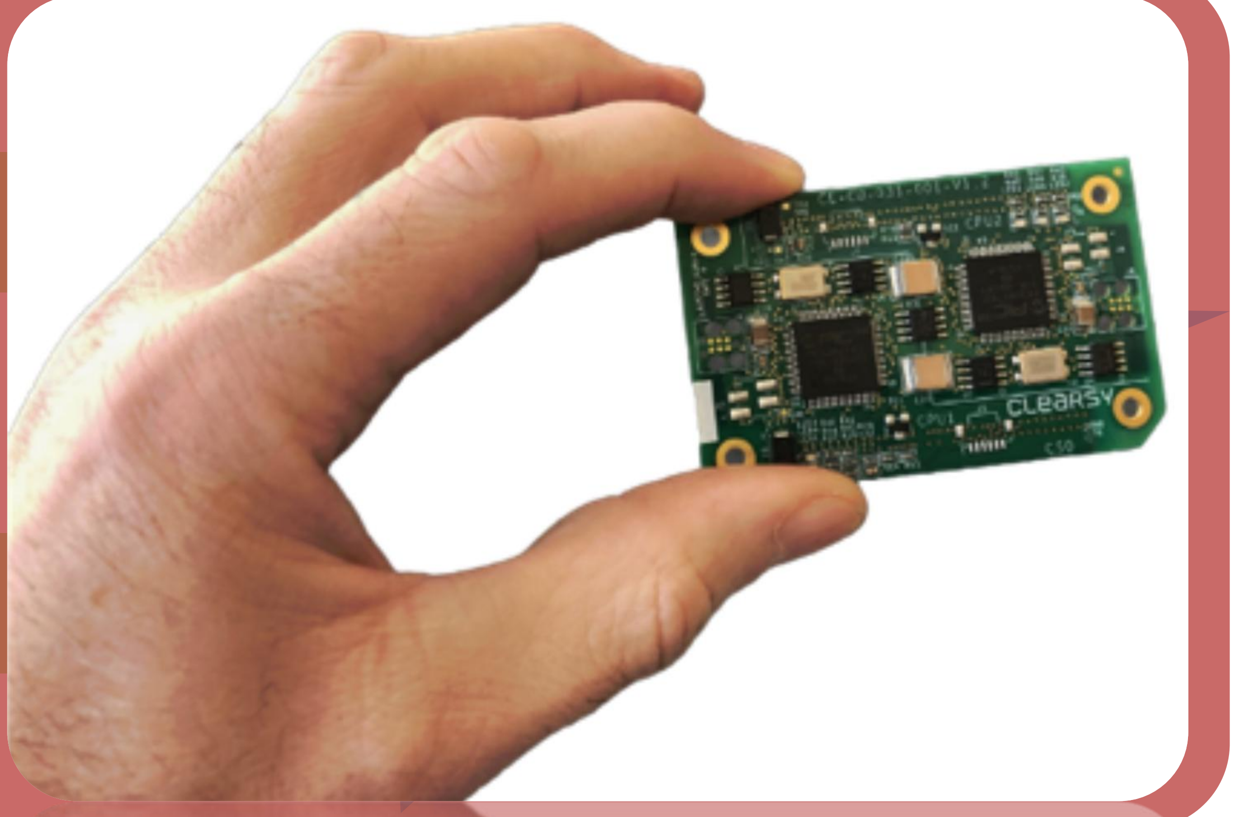
Wrong environment specification

Wrong program

Wrong exploitation procedure

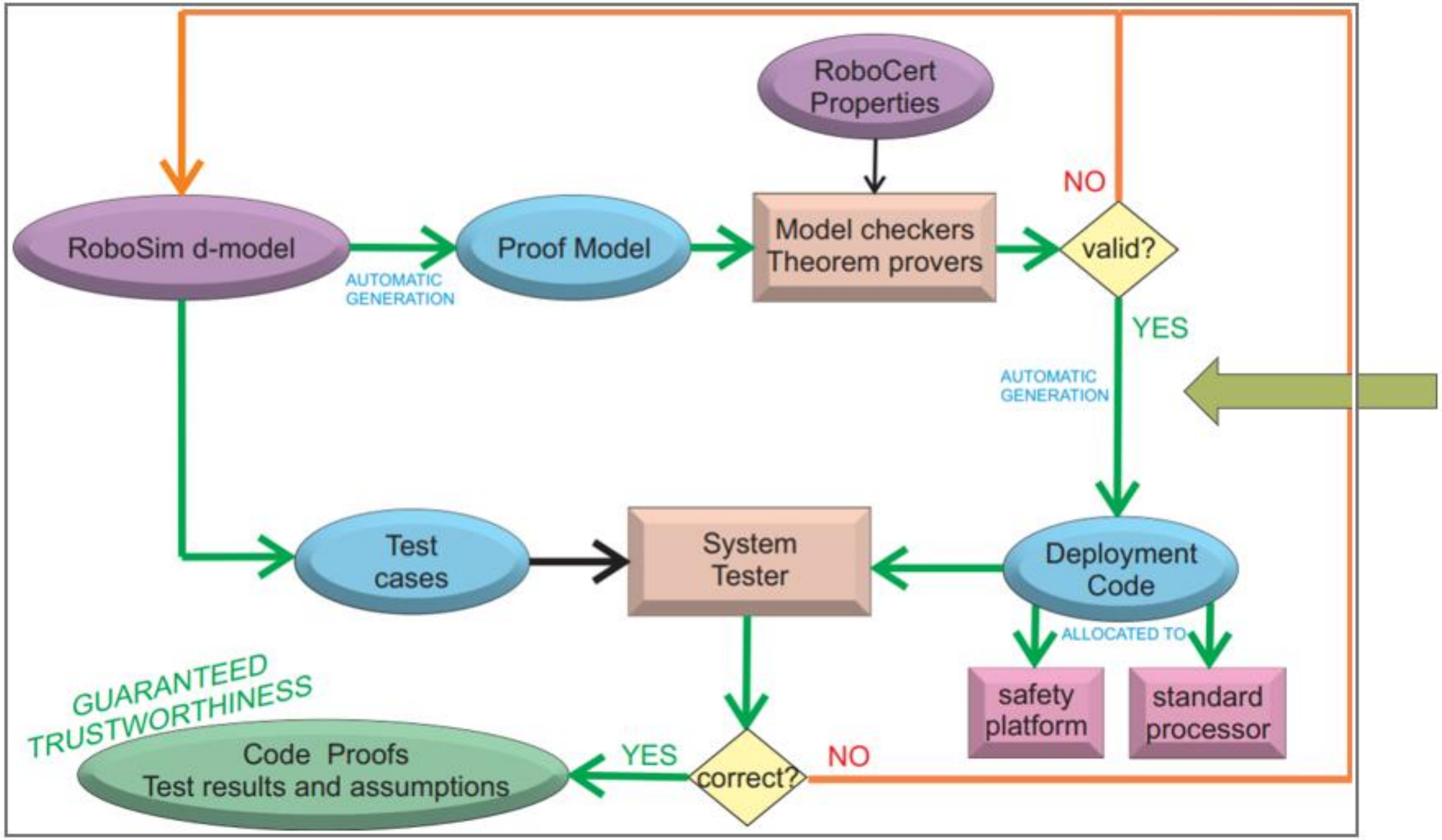
Wrong binary

Wrong execution



ROBOSTAR

The process



Functional Model and Safety Model with Robosim

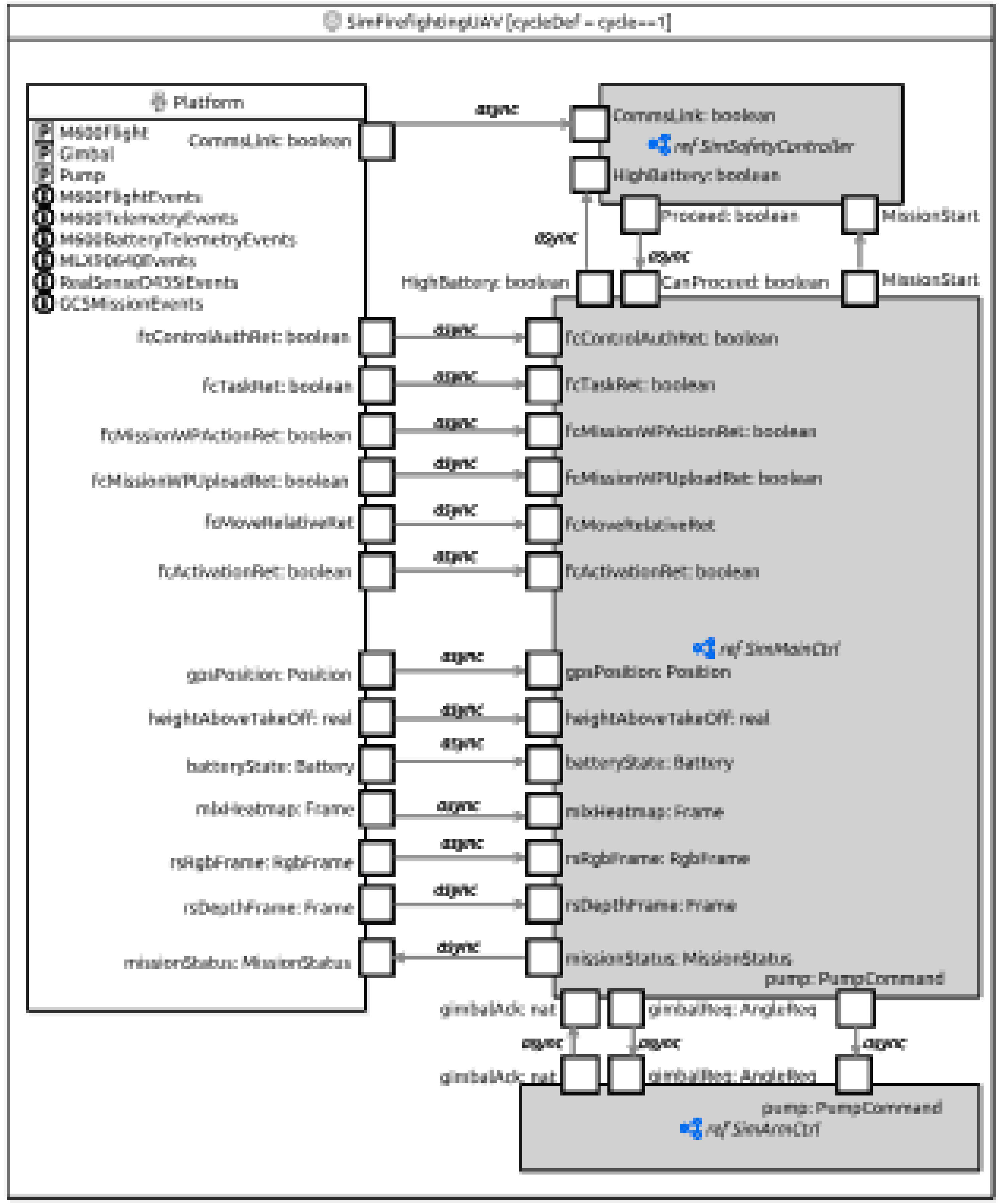
Implementation in B

Execution by CLEARSY Safety Platform

ROBOSTAR

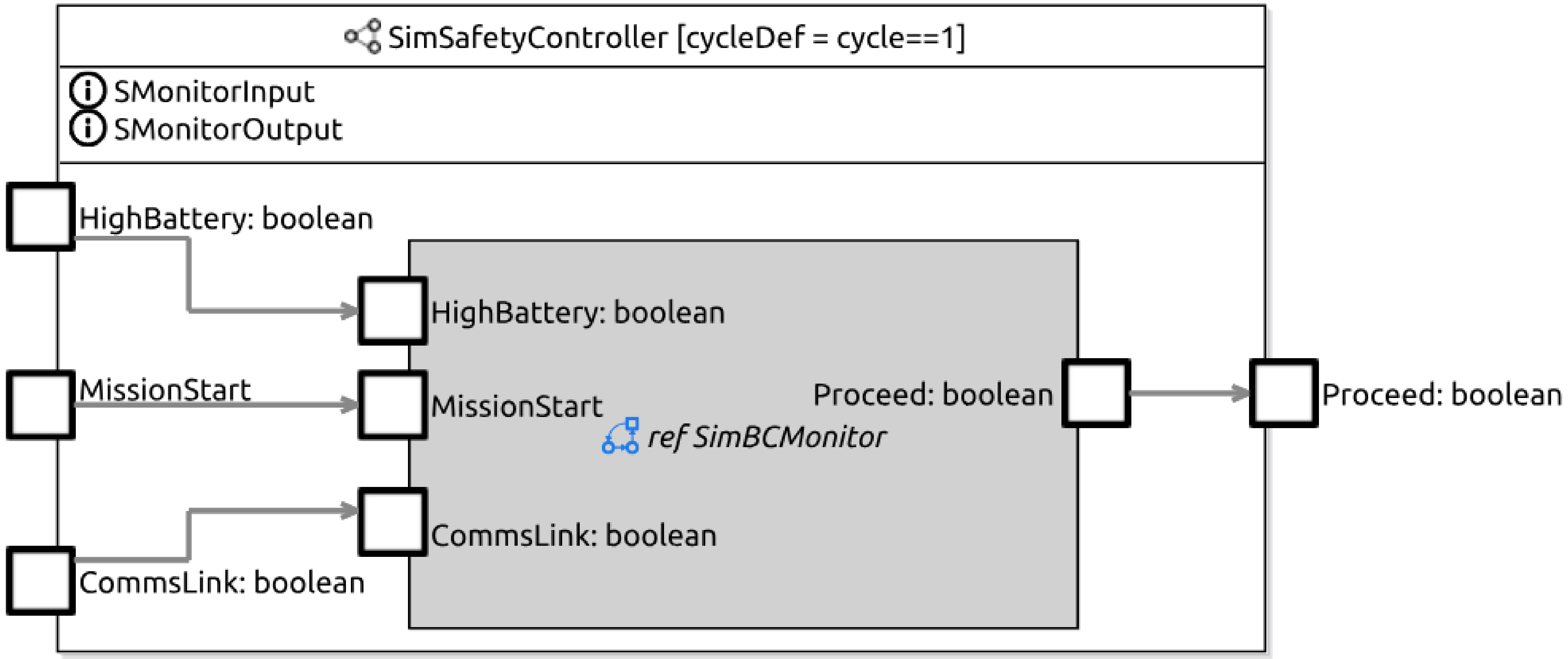
Functional Model and Safety Model with Robosim

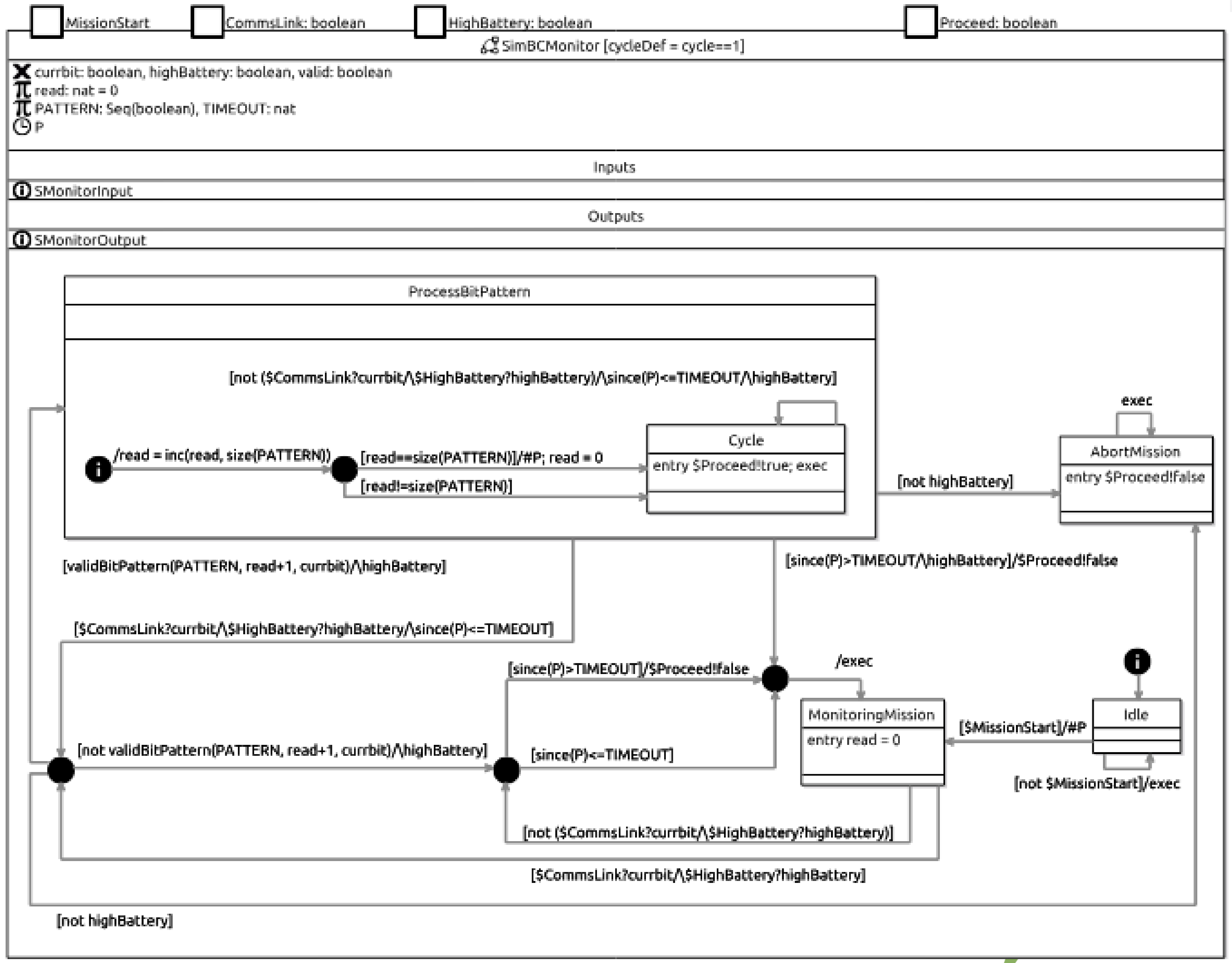
- From NL requirements to Robosim model



- Robotic platform
- Three controllers:
  - SimSafetyController
  - SimMainCtrl
  - SimArmCtrl

Functional Model and Safety Model with Robosim



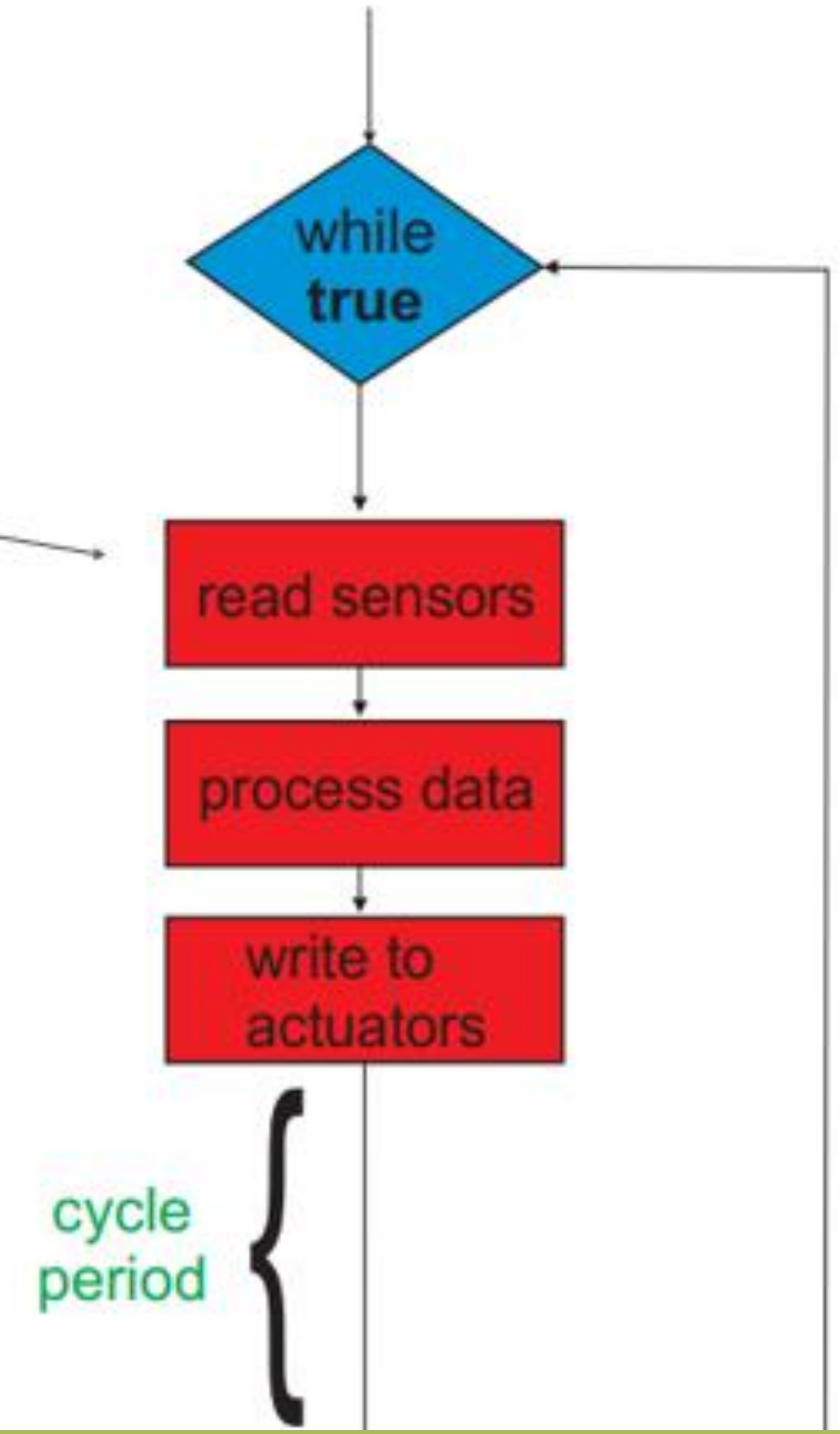


ROBOSTAR

Implementation in B

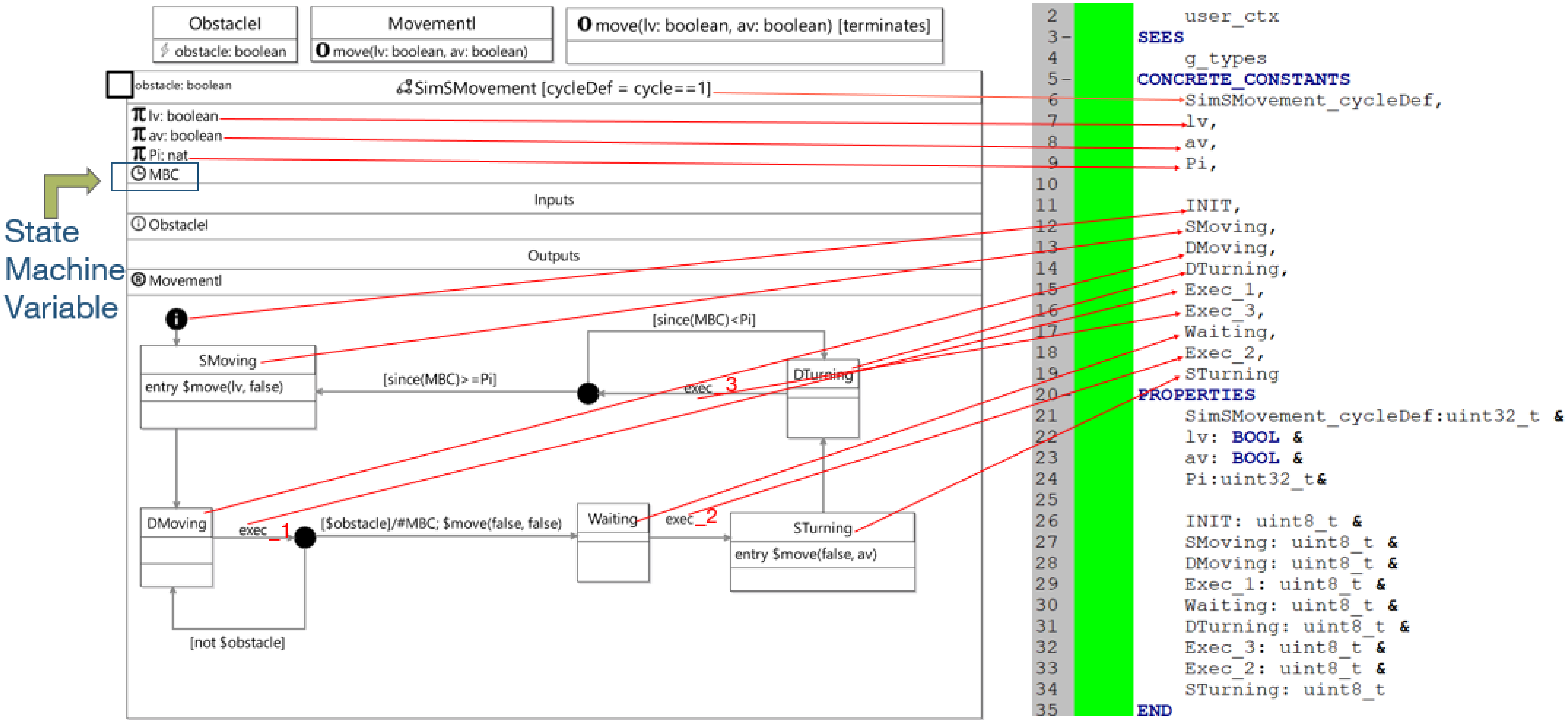
- Robosim and CLEARSY Safety Platform have a common notion of cycle
- Loop: inputs acquisition, computing, outputs control

```
user_logic = BEGIN
    read_master_clock;

    IF first_time = TRUE THEN
        execute_model_cycle;
        first_time := FALSE
    ELSE
        VAR since_value IN
            since_value:(since_value:uint32_t);

            since_value <-- since(SimSMovement_cycle_timer);

            IF since_value < SimSMovement_cycleDef THEN
                skip
            ELSE
                execute_model_cycle
            END
        END
    END
END;
```



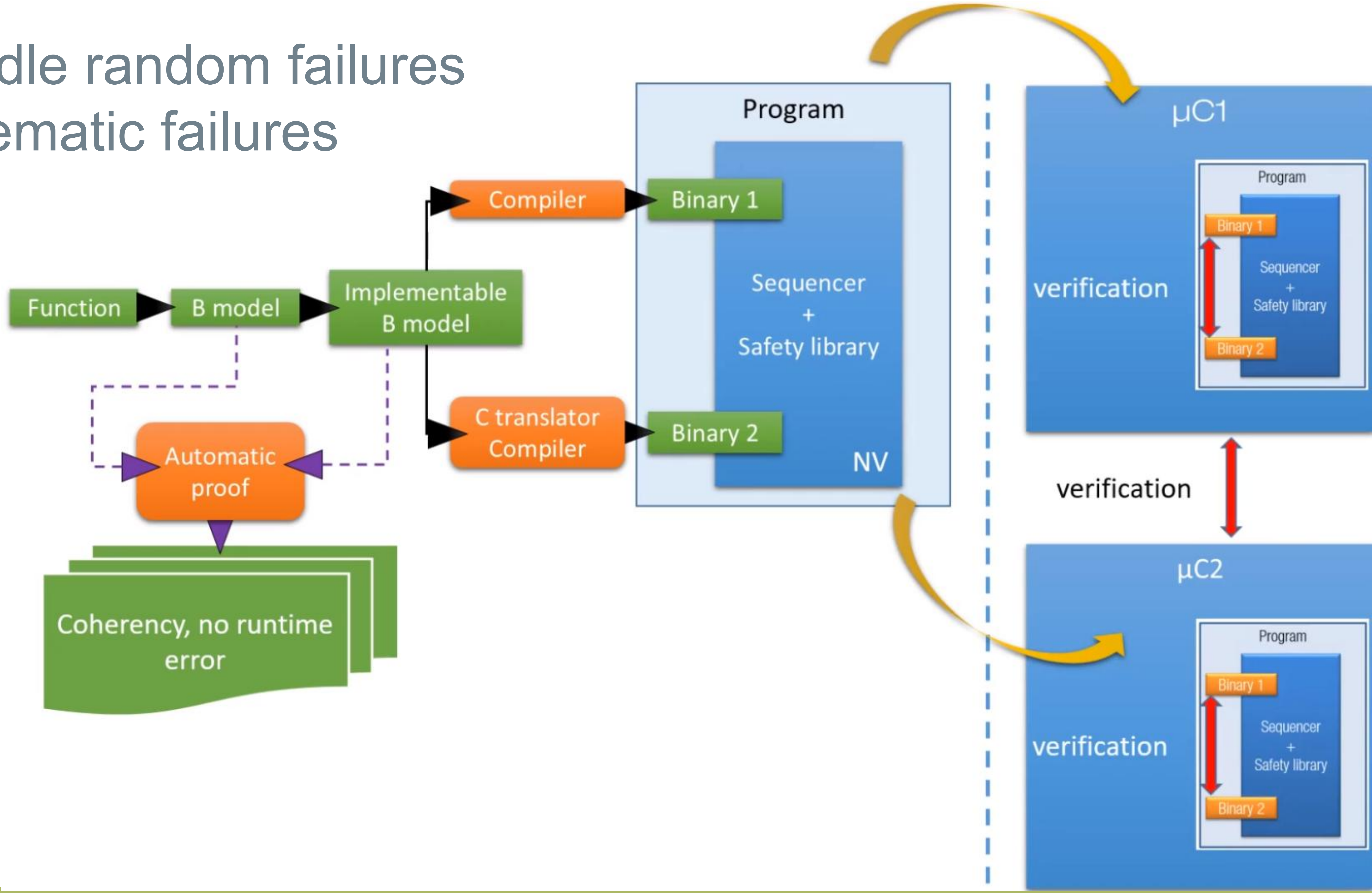
Implementation in B

- Transformation of Robosim into B, as supported by the CLEARSY Safety Platform
- Systematic translation



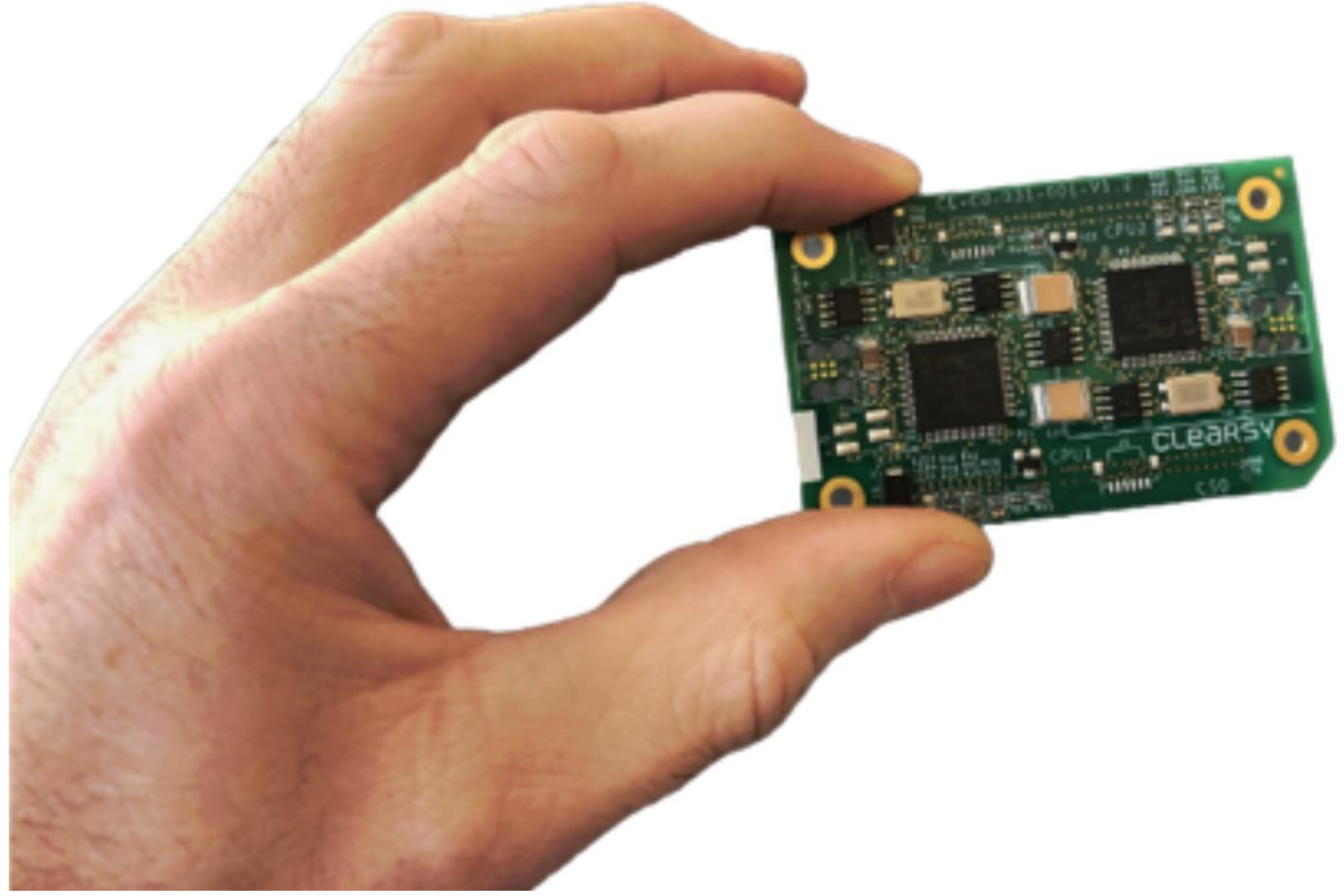
Execution by CLEARSY Safety Platform

- Safety computer able to handle random failures
- Programmed with B for systematic failures

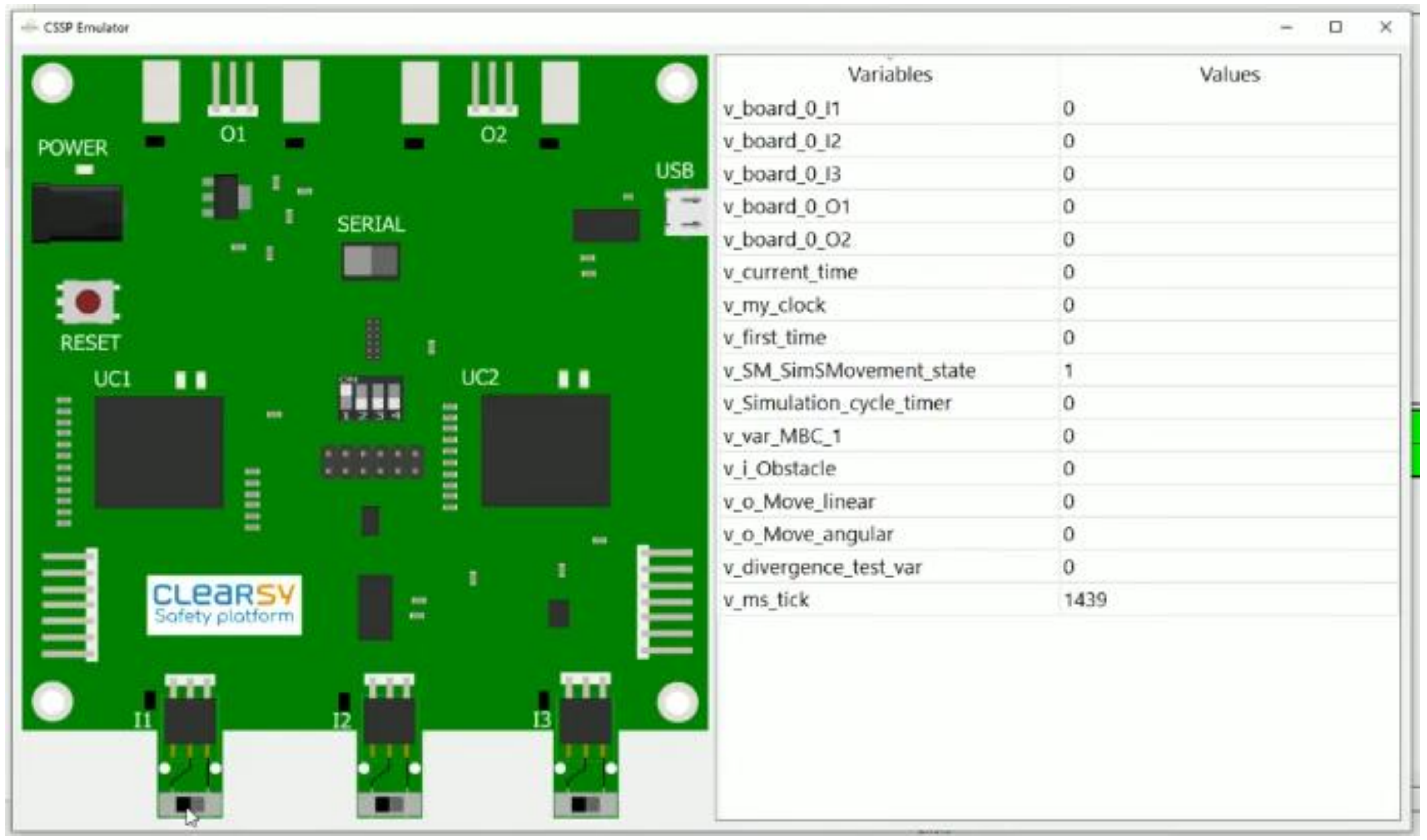


Execution by CLEARSY Safety Platform

Academic board



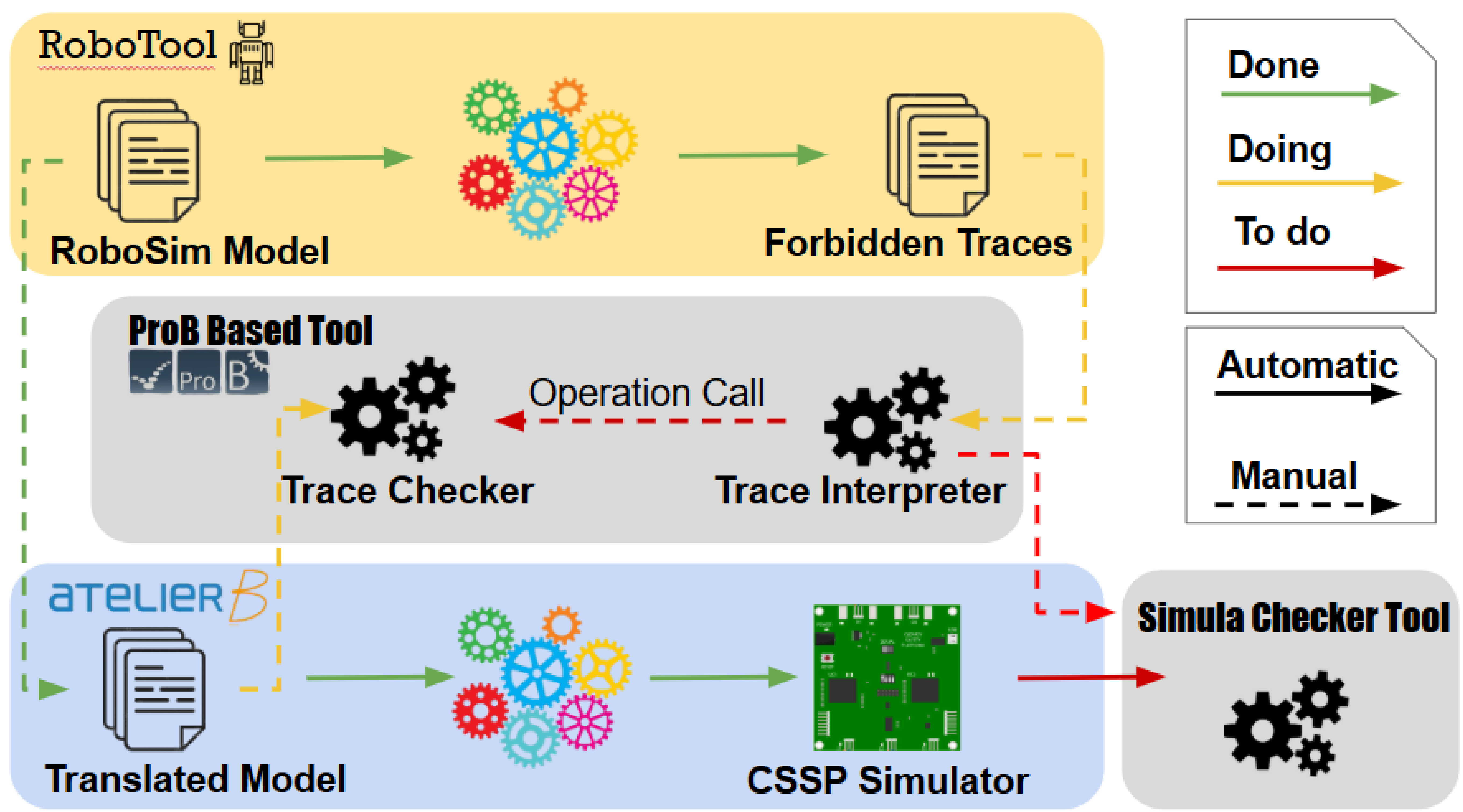
Industrial board



Software simulator

ROBOSTAR

Status



Motivation for 100 students
Summer school
Fortaleza March 2025
https://rome.gesaduece.com.br/

On going research
Verified translation from Robosim to B
UFRN, Brazil

ROBOSTAR

Status

- **Feasible**
- **Verified translation**
  - Demonstration with software simulator
  - Demonstration with academic board
  - Demonstration with industrial board
- **Other (robotics) application**
- **Another summer school in Brasilia (2026)**