CLEARSY
Safety Solutions Designer

Hackathon
JUL2025

AIX
LYON
PARIS
STRASBOURG

WWW.CLEARSY.COM

Airlock
Implementing the Functional Controller

Thierry Lecomte
R&D Director

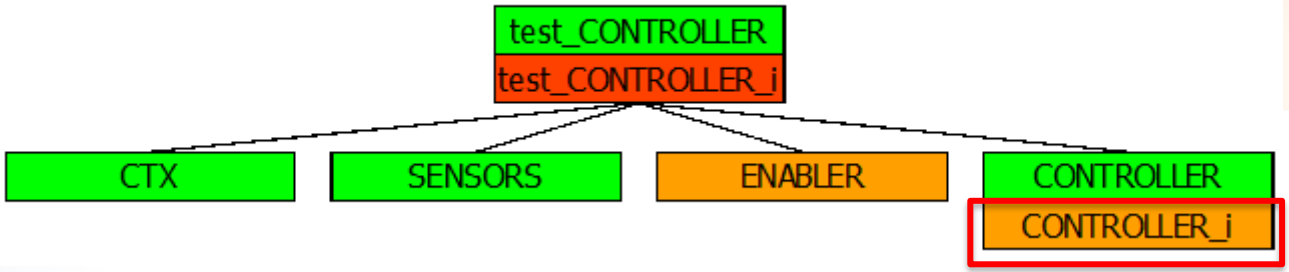
PART IV

One Solution

THIERRY.LECOMTE@CLEARSY.COM

Your turn

- *Implement CONTROLLER [8 pt]*
  - *VARIABLES are already concrete. Only INITIALISATION is required.*
  - *OPERATION process_headers implementation is given*
  - *Implement the algorithm of the OPERATION control*



```
control = /* courtesy of henrique */
BEGIN
    IF current_objective = OBJ_OPEN_DOOR_A THEN
        IF (pressure_sensor_l = PRESSURE_A &
                contact_sensor_b = TRUE &
                contact_sensor_a = FALSE
            )
        THEN
            current_objective := OBJ_NONE; current_action := NONE
        ELSIF contact_sensor_b = FALSE THEN
            current_action := TRANSLATE_CLOSE_DOOR_B
        ELSIF contact_sensor_a = FALSE THEN
            current_action := TRANSLATE_CLOSE_DOOR_A
        ELSIF pressure_sensor_l /= PRESSURE_A THEN
            current_action := ADAPT_PRESSURE_L_TO_A
        ELSE
            current_action := TRANSLATE_OPEN_DOOR_A
        END
```

```
ELSIF current_objective = OBJ_OPEN_DOOR_B THEN
    IF (pressure_sensor_l = PRESSURE_B &
            contact_sensor_b = FALSE &
            contact_sensor_a = TRUE
        )
    THEN
        current_objective := OBJ_NONE; current_action := NONE
    ELSIF contact_sensor_a = FALSE THEN
        current_action := TRANSLATE_CLOSE_DOOR_A
    ELSIF contact_sensor_b = FALSE THEN
        current_action := TRANSLATE_CLOSE_DOOR_B
    ELSIF pressure_sensor_l /= PRESSURE_B THEN
        current_action := ADAPT_PRESSURE_L_TO_B
    ELSE
        current_action := TRANSLATE_OPEN_DOOR_B
    END
```

```
ELSE
    IF (
            (current_authentication = AUTHENTICATED_A &
                button_room_a_open_a = TRUE
            ) or
            (current_authentication = AUTHENTICATED_L &
                button_room_l_open_a = TRUE
            )
        )
    THEN
        current_objective := OBJ_OPEN_DOOR_A
    ELSIF (
            (current_authentication = AUTHENTICATED_B &
                button_room_b_open_b = TRUE
            ) or
            (current_authentication = AUTHENTICATED_L &
                button_room_l_open_b = TRUE
            )
        )
    THEN
        current objective := OBJ OPEN DOOR B
```

```
ELSE
    IF contact_sensor_a = FALSE THEN
        current_action := TRANSLATE_CLOSE_DOOR_A
    ELSIF contact_sensor_b = FALSE THEN
        current_action := TRANSLATE_CLOSE_DOOR_B
    ELSE
        current_action := NONE
    END
END
```

Your turn

- *Complement test_CONTROLLER [2 pt]*
  - *Specify an invariant that links some variables of CONTROLLER and the variables enable_door_a and enable_door_b*

```
INVARIANT
    ( current_action = TRANSLATE_OPEN_DOOR_A => enable_door_a = TRUE ) &
    ( current_action = TRANSLATE_OPEN_DOOR_B => enable_door_b = TRUE ) &
    ( current_action = ADAPT_PRESSURE_L_TO_A => enable_door_a = FALSE ) &
    ( current_action = ADAPT_PRESSURE_L_TO_B => enable_door_b = FALSE )
```

Final status

Component	TypeChecked	POs Generated	Proof Obligations	Proved	Unproved	B0 Checked
ACCESS_CARD	OK	OK	8	6	2	-
CONTROLLER	OK	OK	0	0	0	-
CONTROLLER_i	OK	OK	128	128	0	-
CTX	OK	OK	0	0	0	-
ENABLER	OK	OK	1	1	0	-
SENSORS	OK	OK	0	0	0	-
test_CONTROLLER	OK	OK	0	0	0	-
test_CONTROLLER_i	OK	OK	6	6	0	-

Your turn

- ***Optional**: Have a look at the proof obligation issued from the added invariant in test_CONTROLLER_i and try to **briefly** check if the predicate is valid. Explain **shortly** your thought ? [3 pt]*

- *Missing information to relate sensors, pressure, enabling and disabling*
- *Adaptation of ENABLER compute_enabling post-condition required*

Your turn

```
compute_enabling =
PRE
    (not(pressure_sensor_1 = PRESSURE_A) => contact_sensor_a = TRUE) &
    (not(pressure_sensor_1 = PRESSURE_B) => contact_sensor_b = TRUE)
THEN
    enable_door_a,
    enable_door_b :(
        enable_door_a : BOOL &
        enable_door_b : BOOL &
        (pressure_sensor_1 = PRESSURE_B & contact_sensor_a = TRUE =>
            enable_door_a = FALSE & enable_door_b = TRUE) &
        (pressure_sensor_1 = PRESSURE_A & contact_sensor_b = TRUE =>
            enable_door_a = TRUE & enable_door_b = FALSE) &
        (not(pressure_sensor_1 = PRESSURE_B & contact_sensor_a = TRUE) &
            not(pressure_sensor_1 = PRESSURE_A & contact_sensor_b = TRUE) =>
            enable_door_a = FALSE & enable_door_b = FALSE)
    )
```