Annexe A
Train d'atterrissage d'un avion

Le cahier des charges de cette étude de cas a été proposé dans la conférence ABZ 2014 [Boniol and Wiels, 2014]. Notre objectif est de réaliser le développement du système constitué des besoins, leurs spécifications formelles en B événementiel et les liens entre eux représentés par un glossaire. Le développement a été réalisé sous la plateforme Rodin (version 3.2.0, http://rodin-b-sharp.sourceforge.net/) avec les plugins ProR, ProB, Project Diagram et Event-B Statemachines.

A.1 Quelques exigences du cahier des charges du client

The landing system is in charge of maneuvering landing gears and associated doors. The landing system is composed of 3 landing sets: front, left and right. Each landing set contains a door, a landing gear and associated hydraulic cylinders (...)

The system is controlled digitally in nominal mode and analogically in emergency mode. In this case study, we do not consider the emergency mode. However, in order to allow the pilot to activate the emergency command, the system has to elaborate health parameters for all the equipments involved in the landing gear function. This health monitoring part is in the scope of the case study. In nominal mode, the landing sequence is: open the doors of the landing gear boxes, extend the landing gears and close the doors. After taking off, the retraction sequence to be performed is: open the doors, retract the landing gears and close the doors. (...)

To command the retraction and outgoing of gears, an Up/Down handle is provided to the pilot. When the handle is switched to "Up" the retracting landing gear sequence is executed, when the handle is switched to "Down" the extending landing gear sequence is executed.
The pilot has a set of lights giving the current position of gears and doors, and the current health state of the system and its equipments. These lights are:
- one green light: "gears are locked down",
- one orange light: "gears maneuvering",
- one red light: "landing gear system failure" (...)

Gear cylinders. *Gear cylinders are locked in high or down position by means of a latching box mechanism (the latching boxes are physically on the gears, one for each position). When a gear cylinder is locked in high (resp. down) position and when it receives pressure from the high (resp. down) hydraulic circuit,*
- first it is unlocked from the high (resp. down) position
- then it moves to the down (resp. high) position
- and finally it is locked in the down (resp. high) position. (...)

Door cylinders. *Door cylinders are locked (by means of two latching boxes on each door) only in closed position. Doors are maintained open by maintaining pressure in extension circuit. When a door cylinder is locked in closed position and when it receives pressure from the extension hydraulic circuit,*
- first it is unlocked from the closed position
- then it moves to the open position
- and finally it is maintained in the open position as long as the pressure is maintained in the hydraulic extension circuit.

Outgoing sequence. *The outgoing of gears is decomposed in a sequence of elementary actions. When the gears are locked in retracted position, and the doors are locked in closed position, if the pilot sets the handle to "Down", then the software should have the following sequence of actions:*

1. *stimulate the general electro-valve isolating the command unit in order to send hydraulic pressure to the maneuvering electro-valves,*
2. *stimulate the door opening electro-valve,*
3. *once the three doors are in the open position, stimulate the gear outgoing electro-valve,*
4. *once the three gears are locked down, stop the stimulation of the gear outgoing electro- valve,*
5. *stop the stimulation of the door opening electro-valve,*
6. *stimulate the door closure electro-valve,*
7. *once the three doors are locked in the closed position, stop the stimulation of the door closure electro-valve,*
8. *and finally stop stimulating the general electro-valve.*

(R_{11}bis) *When the command line is working (normal mode), if the landing gear command handle has been pushed DOWN and stays DOWN, then eventually the gears will be locked down and the doors will be seen closed;*

(R_{12}bis) *When the command line is working (normal mode), if the landing gear command handle has been pushed UP and stays UP, then eventually the gears will be locked retracted and the doors will be seen closed.*

A.2 Structuration des besoins à l'aide de ProR

CdC				
ID	*Description*	*Term_ Type*	*Terms*	*Event-B Model*
FUN-G	The landing system goal is maneuvering landing gears and their associated doors	Functionality	- maneuvering	
		Fact	- gears - doors	
EQU-1	The landing system is composed of three sets: front, left and right	Fact	three landing sets: - front - left - right	
EQU-1-1	Each landing set contains: a door, a landing gear and their associated hydraulic cylinders	Fact	a landing set: a door, a landing gear, hydraulic cylinders	
FUN-1	Maneuvering gears consists on extending or retracting them or reversing their movement	Functionality	- extend gears - retract gears - reverse gears movement	
FUN-2	Maneuvering doors consists on opening or closing them	Functionality	- open doors - close doors	
FUN-3	The doors must be open when extending or retracting gears	Obligation	*pre-condition:* - doors must be open *post-condition:* - extend gears - retract gears	
FUN-Ctrl	There are 2 controlling ways of the landing system: digitally or analogically	Fact	controlling ways: - digital - analogical	
FUN-Modes	Landing system state can be either in nominal mode or in emergency mode	Fact	operating modes: - nominal - emergency	
FUN-Digi	When the system is in nominal mode, it is controlled digitally	Obligation	nominal mode ⇒ digital control	
FUN-Ana	When the system is in emergency mode it is controlled analogically	Obligation	emergency mode ⇒ analogical control	
FUN-Hlth	The system must elaborate health parameters for all equipments involved in the landing gear function according to them the pilot can activate emergency command	Fact	- health parameters	
		Functionality	- elaborate health parameters - pilot activates emergency command	
LS	In nominal mode, the landing sequence is: open the doors → extend the landing gears → close the doors	Behavior	landing sequence: open doors → extend gears → close doors	
RS	In nominal mode, the retracting sequence is: open the doors → retract the landing gears → close the doors	Behavior	retracting sequence: open doors → retract gears → close doors	

FUN-P-I	The pilot interface is composed of green, orange and red lights and an UP/DOWN handle	Fact	- green light - orange light - red light - an UP/DOWN handle	
FUN-P-handle1	When handle is UP, landing gears are retracted	Obligation	*pre-condition:* handle is UP *post-condition:* gears are retracted	
FUN-P-handle2	When handle is DOWN, landing gear are extended	Obligation	*pre-condition:* handle is DOWN *post-condition:* gears are extended	
FUN-P-light1	When gears are locked down, the green light is on	Obligation	*pre-condition:* gears are locked down *post-condition:* green light = on	
FUN-P-light2	When gears are maneuvering, the orange light is on	Obligation	*pre-condition:* gears are maneuvering *post-condition:* orange light = on	
FUN-P-light3	When the landing gears system is failed, the red light is on	Obligation	*pre-condition:* system failed *post-condition:* red light = on	
FUN-P-light4	Every light can be "on" or "off"	Fact	- on - off	
FUN-P-light5	All lights are "off" when gears are locked up	Obligation	*pre-condition:* gears are locked up *post-condition:* green light = off & orange light = off & red light = off	
EQU-LBx	Gear cylinders are locked in high or down position by means of a latching box	Fact	- gears cylinders - locked in high position - latching box	
EQU-LBx1	A latching box is physically on the gears	Fact	latching box on the gears	
EQU-LBx2	There is one latching box for each position: one for high position and one for down position	Fact	- one latching box for high position - one latching box for down position	
GCy	When a gear cylinder is locked in high position, and when it receives pressure from high hydraulic circuit, - first it is unlocked from the high position - then it moves to the down position - and finally it is locked in the down position.	Obligation Behavior	*pre-condition:* - a gear cylinder locked in high position - it receives pressure from high hydraulic circuit *post-condition:* unlock gear cylinder from high position → move gear cylinder to the down position → lock gear cylinder in the down position	
GCy-dwn	When a gear cylinder is locked in down position, and when it receives pressure from down hydraulic circuit, - first it is unlocked from the down position - then it moves to the high position - and finally it is locked in the high position.	Obligation Behavior	*pre-condition:* - a gear cylinder is locked in down position - it receives pressure from down hydraulic circuit *post-condition:* unlock gear cylinder from down position → move gear cylinder to the high position → lock gear cylinder in the high position	
EQU-DCy1	Door cylinders are locked only in closed position by means of 2 latching boxes (2 latching boxes in each door)	Fact	- door cylinders - locked only in closed position - 2 latching box in each door	
EQU-DCy2	Doors are maintained open by maintaining pressure in the extension circuit	Fact	pressure in the extension circuit	
DCy	When a door cylinder is locked in closed position and when it receives pressure from the extension hydraulic circuit,	Obligation Behavior	*pre-condition:* - a door cylinder is locked in closed position	

	- first it is unlocked from the closed position - then it moves to the open position - and finally it is maintained in the open position as long as the pressure is maintained in the hydraulic extension circuit		- it receives pressure from the extension hydraulic circuit *post-condition:* unlock door cylinder from closed position → move door cylinder to the open position → lock door cylinder in the open position	
GEv	When the gears are locked in retracted position, and the doors are locked in closed position, if the pilot sets the handle to "Down", then the software should have the following sequence of actions: 1. stimulate the general electro-valve isolating the command unit in order to send hydraulic pressure to the maneuvering electro- valves, 2. stimulate the door opening electro-valve, 3. once the three doors are in the open position, stimulate the gear outgoing electro-valve, 4. once the three gears are locked down, stop the stimulation of the gear outgoing electro-valve, 5. stop the stimulation of the door opening electro-valve, 6. stimulate the door closure electro-valve, 7. once the three doors are locked in the closed position, stop the stimulation of the door closure electro-valve, 8. and finally stop stimulating the general electro-valve	Obligation Behavior	*pre-condition:* - the gears are locked in retracted position - the doors are locked in closed position - the pilot sets the handle to "Down" *post-condition:* stimulate the general electro-valve → stimulate the door opening electro-valve → stimulate the gear outgoing electro-valve → stop the stimulation of the gear outgoing electro-valve → stop the stimulation of the door opening electro-valve → stimulate the door closure electro-valve → stop the stimulation of the door closure electro-valve → stop stimulating the general electro-valve	
R_{11}bis	When the command line is working (normal mode), if the landing gear command handle has been pushed DOWN and stays DOWN, then eventually the gears will be locked down and the doors will be seen closed	Obligation Behavior	*pre-condition:* - normal mode - the landing gear command handle has been pushed DOWN - the landing gear command handle stays DOWN *post-condition:* - the gears will be locked down - the doors will be seen closed	
R_{12}bis	When the command line is working (normal mode), if the landing gear command handle has been pushed UP and stays UP, then eventually the gears will be locked retracted and the doors will be seen closed	Obligation Behavior	*pre-condition:* - normal mode - the landing gear command handle has been pushed UP - the landing gear command handle stays UP *post-condition:* - the gears will be locked retracted - the doors will be seen closed	

NB.

1) Les besoins *FUN-1*, *FUN-2* et *FUN-3* ne sont pas explicitement mentionnés dans le cahier des charges du client. Ils sont issus de notre compréhension.

2) Les besoins $R_{11}bis$ et $R_{12}bis$ sont décrits dans la partie du cahier des charges intitulée *« Requirements / Properties »* décrite par l'informaticien. Les autres besoins proviennent de la partie explicative du même document.

A.3 Quelques étapes de développement

Ce développement est guidé par les besoins : nous partons de notre compréhension du CdC et élaborons la Spec formelle. Le glossaire est mis à jour automatiquement.

A.3.1 Initialisation de la Spec

CdC			
ID	*Description*	*Terms*	*Event-B Model*
FUN-G	The **[Landing_System]** goal is maneuvering landing gears and their associated doors	- maneuvering - gears - doors	**Landing_Ctx, Landing_System**
EQU-1	The **[Landing_System]** is composed of three sets: front, left and right	three landing sets: - front - left - right	**Landing_Ctx, Landing_System**
EQU-1-1	Each landing set contains: a door, a landing gear and their associated hydraulic cylinders	a landing set: a door, a landing gear, hydraulic cylinders	
FUN-1	Maneuvering gears consists on extending or retracting them or reversing their movement	- extend gears - retract gears - reverse gears movement	
FUN-2	Maneuvering doors consists on opening or closing them	- open doors - close doors	
FUN-3	The doors must be open when extending or retracting gears	*pre-condition:* - doors must be open *post-condition:* - extend gears - retract gears	
FUN-Ctrl	There are 2 controlling ways of the **[Landing_System]**: digitally or analogically	controlling ways: - digital - analogical	**Landing_Ctx, Landing_System**
FUN-Modes	**[Landing_System]** state can be either in nominal mode or in emergency mode	operating modes: - nominal - emergency	**Landing_Ctx, Landing_System**
LS	In nominal mode, the landing sequence is: open the doors → extend the landing gears → close the doors	landing sequence: open doors → extend gears → close doors	

Glossaire	
Formal term	*Informal description*
Landing_System	**landing system**

Spec

CONTEXT **Landing_Ctx**
SETS
CONSTANTS
AXIOMS
END

MACHINE **Landing_System**
SEES **Landing_Ctx**
VARIABLES
INVARIANTS
EVENTS
INITIALISATION
END

GCy	When a gear cylinder is locked in high position, and when it receives pressure from high hydraulic circuit, - first it is unlocked from the high position - then it moves to the down position - and finally it is locked in the down position.	*pre-condition:* - a gear cylinder locked in high position - it receives pressure from high hydraulic circuit *post-condition:* unlock gear cylinder from high position → move gear cylinder to the down position → lock gear cylinder in the down position	
DCy	When a door cylinder is locked in closed position and when it receives pressure from the extension hydraulic circuit, - first it is unlocked from the closed position - then it moves to the open position - and finally it is maintained in the open position as long as the pressure is maintained in the hydraulic extension circuit	*pre-condition:* - a door cylinder is locked in closed position - it receives pressure from the extension hydraulic circuit *post-condition:* unlock door cylinder from closed position → move door cylinder to the open position → lock door cylinder in the open position	
...	...	...	

A.3.2 Introduction des modes nominal et d'urgence

CdC

ID	*Description*	*Terms*	*Event-B Model*
FUN-G	The [Landing_System] goal is maneuvering landing gears and their associated doors	- maneuvering - gears - doors	Landing_Ctx, Landing_System
EQU-1	The [Landing_System] is composed of three sets: front, left and right	three landing sets: - front - left - right	Landing_Ctx, Landing_System
EQU-1-1	Each landing set contains: a door, a landing gear and their associated hydraulic cylinders	a landing set: a door, a landing gear, hydraulic cylinders	
FUN-1	Maneuvering gears consists on extending or retracting them or reversing their movement	- extend gears - retract gears - reverse gears movement	
FUN-2	Maneuvering doors consists on opening or closing them	- open doors - close doors	
FUN-3	The doors must be open when extending or retracting gears	*pre-condition:* - doors must be open *post-condition:* - extend gears - retract gears	
FUN-Ctrl	There are 2 controlling ways of the [Landing_System]: digitally or analogically	controlling ways: - digital - analogical	Landing_Ctx, Landing_System
FUN-Modes	[Landing_System] state can be either in **[nominal] [op_mode]** or in **[emergency] [op_mode]**	**[op_mode]**: - **[nominal]** - **[emergency]**	Landing_Ctx, Landing_System
LS	In **[nominal] [op_mode]**, the landing sequence is: open the doors → extend the landing gears → close the doors	landing sequence: open doors → extend gears → close doors	**Landing_Ctx, Landing_System**
GCy	When a gear cylinder is locked in high position, and when it receives pressure from high hydraulic circuit, - first it is unlocked from the high position	*pre-condition:* - a gear cylinder locked in high position	

Glossaire

Formal term	*Informal description*
Landing_System	landing system
nominal	**nominal**
op_mode	**mode, operating modes**
emergency	**emergency**

Spec

CONTEXT Landing_Ctx
SETS
  Mode
CONSTANTS
  nominal, emergency
AXIOMS
  axm_env: partition(Mode, {nominal}, {emergency})
END

MACHINE Landing_System
SEES Landing_Ctx
VARIABLES
  op_mode
INVARIANTS
  inv_Typ_op_mode : op_mode ∈ Mode
EVENTS
INITIALISATION
  then
    act1: op_mode := nominal
  end
Event **evolve_op_mode**
 any
    m
where
   grd1: m ∈ Mode
 then
   act1: op_mode := m
end
END

	- then it moves to the down position - and finally it is locked in the down position.	- it receives pressure from high hydraulic circuit *post-condition:* unlock gear cylinder from high position → move gear cylinder to the down position → lock gear cylinder in the down position	
DCy	When a door cylinder is locked in closed position and when it receives pressure from the extension hydraulic circuit, - first it is unlocked from the closed position - then it moves to the open position - and finally it is maintained in the open position as long as the pressure is maintained in the hydraulic extension circuit	*pre-condition:* - a door cylinder is locked in closed position - it receives pressure from the extension hydraulic circuit *post-condition:* unlock door cylinder from closed position → move door cylinder to the open position → lock door cylinder in the open position	
...	...	...	

La **validation** est prise en compte au fur et à mesure de l'évolution du développement du système. Les termes informels *« nominal »*, *« emergency »* et *« operating modes »* extraits du besoin FUN-Modes sont formalisés dans la Spec. Ce sont les deux constantes *nominal* et *emergency* du contexte *Landing_Ctx* et la variable *op_mode* de la machine *Landing_System*. Ces éléments formels font partie du glossaire faisant le lien entre le CdC et la Spec.

A.3.3 Introduction des portes et des trains

CdC		
ID	*Description*	*Terms*
FUN-G	The [Landing_System] goal is maneuvering **[gears_pos]** and their associated **[doors_pos]**	- maneuvering - **[gears_pos]** - **[doors_pos]**
EQU-1	The [Landing_System] is composed of three sets: front, left and right	three landing sets: - front - left - right
EQU-1-1	Each landing set contains: a door, a landing gear and their associated hydraulic cylinders	a landing set: a door, a landing gear, hydraulic cylinders
FUN-1	Maneuvering **[gears_pos]** consists on **[extend_gears]** or **[retract_gears]** or **[reverse_gears_mvt]**	- **[extend_gears]** - **[retract_gears]** -**[reverse_gears_mvt]**
FUN-2	Maneuvering **[doors_pos]** consists on **[open_doors]** or **[close_doors]**	- **[open_doors]** - close doors
FUN-3	The **[doors_pos]** must be **[open]** when **[extend_gears]** or **[retract_gears]**	*pre-condition:* - **[doors_pos]** must be **[open]** *post-condition:* - **[extend_gears]** - **[retract_gears]**
FUN-Ctrl	There are 2 controlling ways of the [Landing_System]: digitally or analogically	controlling ways: - digital - analogical
FUN-Modes	[Landing_System] state can be either in [nominal] [op_mode] or in [emergency] [op_mode]	[op_mode]: - [nominal] - [emergency]
LS	In [nominal] [op_mode], the landing sequence is: **[open_doors]** → **[extend_gears]** → **[close_doors]**	landing sequence: **[open_doors]** → **[extend_gears]** → **[close_doors]**
GCy	When a gear cylinder is locked in high position, and when it receives pressure from high hydraulic circuit,	*pre-condition:* - a gear cylinder locked in high position

Glossaire	
Formal term	*Informal description*
Landing_System	landing system
nominal	nominal
op_mode	mode, operating modes
emergency	emergency
gears_pos	**gears, landing gears**
doors_pos	**doors**
extend_gears	**extending, extend the gears, extend the landing gears**
retract_gears	**retracting gears retract the gears**
reverse_gears_mvt	**reverse gears movement**
open_doors	**open the doors, opening, open doors**
close_doors	**close the doors, closing, close doors**
open	**open**
closed	**closed**
g_extended	**extended**
g_retracted	**retracted**

Spec

CONTEXT Landing_Ctx
SETS
  Mode, **D_Positions, G_Positions,**
CONSTANTS
  nominal, emergency
  open, closed, g_extended, g_retracted
AXIOMS
  axm_env: partition(Mode, {nominal}, {emergency})
  axm_1: partition(D_Positions, {open}, {closed})
  axm_2: partition(G_Positions, {g_extended}, {g_retracted})
END

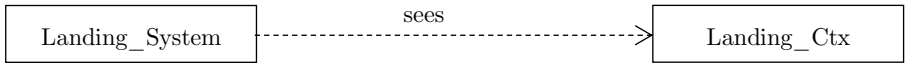
MACHINE Landing_System
SEES Landing_Ctx
VARIABLES
  op_mode, **doors_pos, gears_pos, ext_seq**
INVARIANTS
  inv_Typ_op_mode: op_mode ∈ Mode
  inv_Typ_doors_pos: doors_pos ∈ D_Positions
  inv_Typ_gears_pos: gears_pos ∈ G_Positions
  inv_Typ_ext_seq: ext_seq ∈ BOOL
  inv_1: doors_pos = open ⇒ op_mode = nominal
EVENTS
INITIALISATION
 then
  act1: op_mode := nominal
  act2: doors_pos := closed
  act3: gears_pos := g_retracted
  act3: ext_seq:= FALSE
 end
Event **extend_gears**
 when
  grd1: doors_pos = open
  grd2: gears_pos = g_retracted
  grd2: ext_seq = TRUE
 then
  act1: gears_pos := g_extended
end

	- first it is unlocked from the high position - then it moves to the down position - and finally it is locked in the down position.	- it receives pressure from high hydraulic circuit *post-condition:* unlock gear cylinder from high position → move gear cylinder to the down position → lock gear cylinder in the down position
DCy	When a door cylinder is locked in closed position and when it receives pressure from the extension hydraulic circuit, - first it is unlocked from the closed position - then it moves to the open position - and finally it is maintained in the open position as long as the pressure is maintained in the hydraulic extension circuit	*pre-condition:* - a door cylinder is locked in closed position - it receives pressure from the extension hydraulic circuit *post-condition:* unlock door cylinder from closed position → move door cylinder to the open position → lock door cylinder in the open position
...	...	...

```
Event retract_gears
  when
    grd1: doors_pos = open
    grd2: gears_pos = g_extended
  then
    act1: gears_pos := g_retracted
  end
Event reverse_gears_mvt
  then
    SKIP
  end
Event open_doors
  when
    grd1: op_mode = nominal
    grd2: doors_pos = closed
    grd3: ext_seq = TRUE
  then
    act1: doors_pos := open
  end
Event close_doors
  when
    grd1: doors_pos = open
  then
    act1: doors_pos := closed
  end
Event activate_ext_sequence // event specific to the Spec
  when
    grd1: ext_seq = FALSE
    grd2: doors_pos = closed
  then
    act1: ext_seq := TRUE
  end
Event evolve_op_mode
  any
    m
  where
    grd1: m ∈ Mode
  then
    act1: op  mode := m
  end
END
```

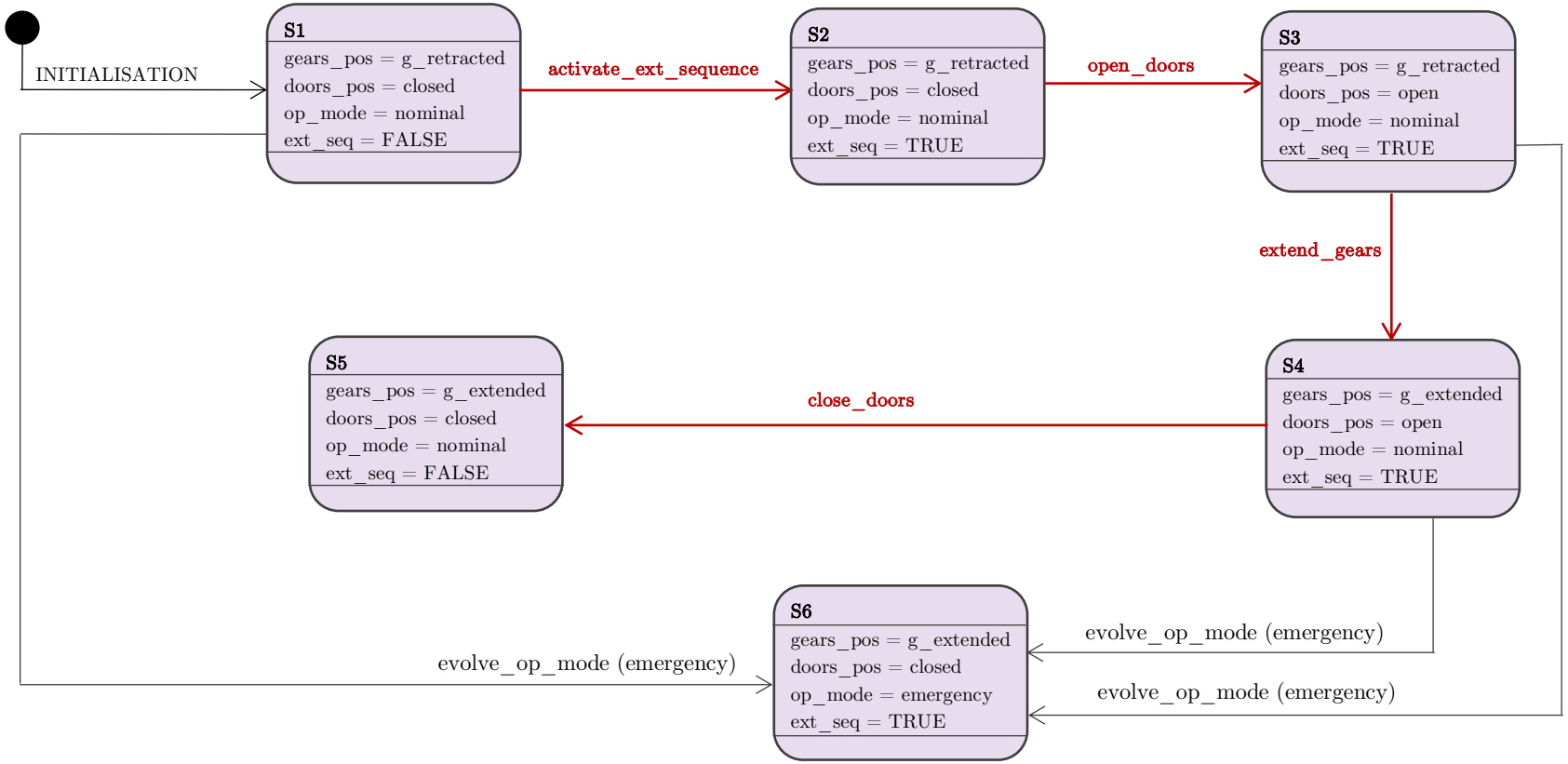
A.3.4 Utilisation des outils

A.3.4.1 Schéma des Specs développées (outil : Project Diagram)



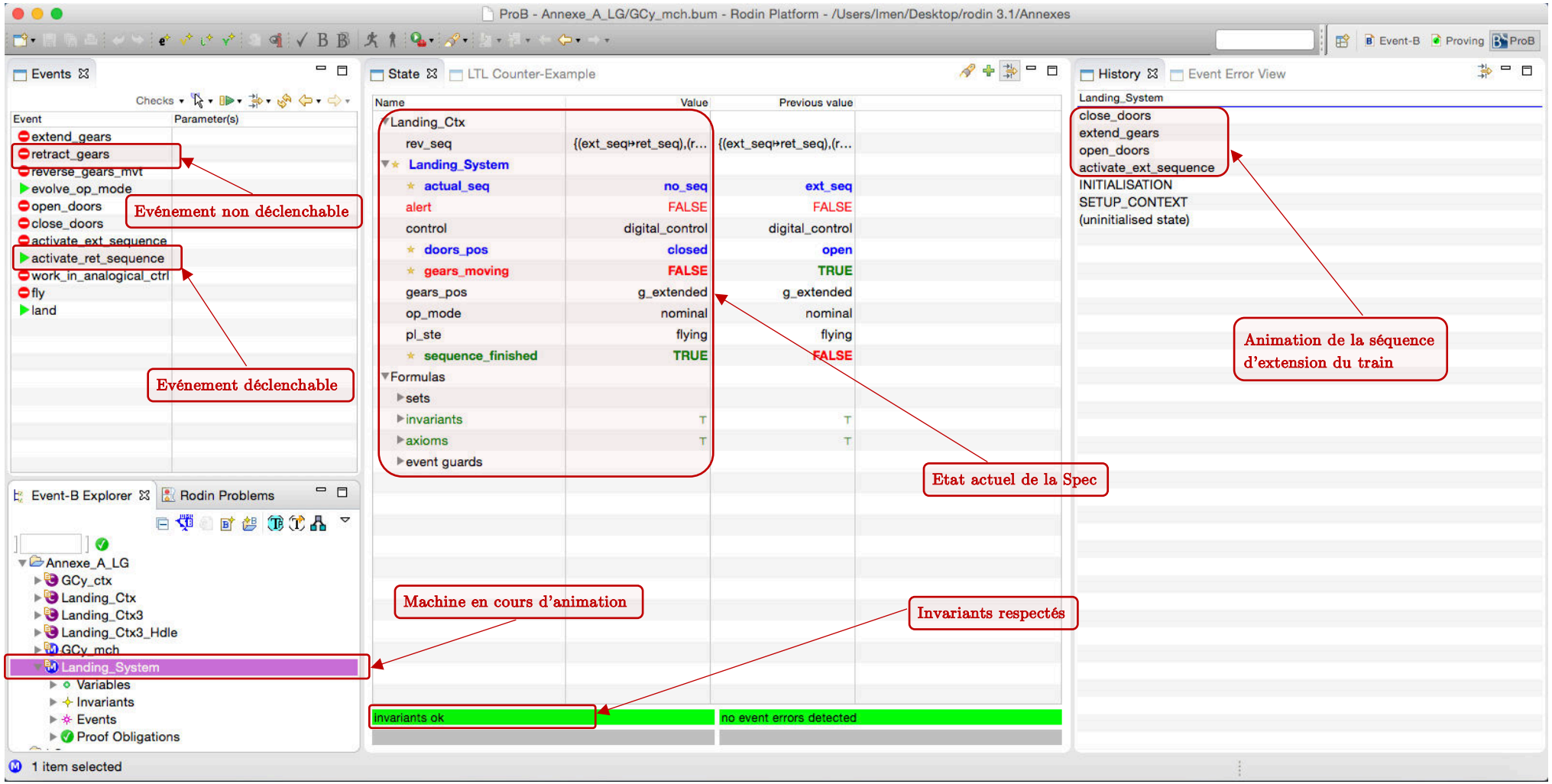
A.3.4.2 Machine d'états-transitions (outil : Event-B Statemachines)

Nous présentons ci-dessous une sous-machine de la machine à états générée automatiquement et associée à la Spec *Landing_System.*
Prenons la séquence d'extension de train décrite par le chemin allant de l'état S1 à l'état S5 en passant par S2, S3 et S4. Ce chemin est décrit par l'enchaînement *activate_ext_sequence* → *open_doors* → *extend_gears* → *close_doors.* Nous introduisons dans la Spec l'événement *activate_ext_sequence* permettant de déclencher la séquence d'extension. Cet événement ne figure pas explicitement dans les besoins du CdC.



A.3.4.3 Validation (outil : ProB)

Animation et model-checking de la Spec en exécutant/déclenchant la séquence d'extension du train décrite dans A.3.4.2



La Spec a exécuté correctement la séquence citée dans A.3.4.2, elle est donc valide relativement à ce comportement. Le model-checker de ProB nous indique également que tous les événements sont respectés pendant l'activité d'animation.

A.3.4.4 Vérification (outils : générateurs d'OPs et prouveurs)

Spec	*OPs automatiques*	*OPs interactives*	*Total*
Landing_Ctx	0	0	0
Landing_System	29	0	29

A.4 Utilisation du patron Dev-seq

Développement guidé par le patron : nous commençons par la réécriture de quelques besoins du CdC afin de préparer à l'instanciation de Dev-seq.

A.4.1 Réécriture des besoins LS, GCy et DCy

Introduction de la notion **d'ordre** dans ces besoins et ajout de quelques termes informels.

CdC	
ID	*Description*
FUN-G	The [Landing_System] goal is maneuvering [gears_pos] and their associated [doors_pos]
...	...
LS	In [nominal] [op_mode], the landing sequence is:
LS-1	**1.** [open_doors]
LS-2	**2.** [extend_gears]
LS-3	**3.** [close_doors]
GCy	When a gear cylinder is locked in high position, and when it receives pressure from high hydraulic circuit, **the extension gears sequence is:**
GCy-1	**1.** unlock the gear cylinder from the high position
GCy-2	**2.** move it to the down position
GCy-3	**3.** lock it in the down position
DCy	When a door cylinder is locked in closed position and when it receives pressure from the extension hydraulic circuit, **the opening doors sequence is:**
DCy-1	**1.** unlock the door cylinder from the closed position
DCy-2	**2.** move it to the open position
DCy-3	**3.** maintain it in the open position as long as the pressure is maintained in the hydraulic extension circuit
...	...

A.4.2 Prise en compte du besoin GCy

Le patron raffine le développement existant de A.3.3 dans lequel LS a été développé. Il génère automatiquement le système suivant :

CdC	
ID	*Description*
FUN-G	The [Landing_System] goal is maneuvering [gears_pos] and their associated [doors_pos]
...	...
LS	In [nominal] [op_mode], the landing sequence is:
LS-1	1. [open_doors]
GCy' // LS-2	**2. When [gears_cyln] is [locked_up], and when it receives [pressure_high], [extend_gears] is:**
GCy'-1	**2.1. [unlock_up_gears_cylinders]**
GCy'-2	**2.2. [move_down_gears_cylinders]**
GCy'-3	**2.3. [lock_down_gears_cylinders]**
LS-3	3. [close_doors]
DCy	When a door cylinder is locked in closed position and when it receives pressure from the extension hydraulic circuit, the opening doors sequence is:
DCy-1	1. unlock the door cylinder from the closed position
DCy-2	2. move it to the open position
DCy-3	3. maintain it in the open position as long as the pressure is maintained in the hydraulic extension circuit

Glossaire	
Formal term	*Informal description*
Landing_System	landing system
nominal	nominal
op_mode	mode, operating modes
emergency	emergency
gears_pos	gears, landing gears
doors_pos	doors
extend_gears	extending, extend the gears, extend the landing gears, **the extension gears sequence**
open_doors	open the doors, opening, open doors
...	...
gears_cyln	**a gear cylinder**
locked_up	**locked in high position**
pressure_high	**pressure from high hydraulic circuit**
unlock_up_gears_cylinders	**unlock the gear cylinder from the high position**
move_down_gears_cylinders	**move it to the down position**
lock_down_gears_cylinders	**lock it in the down position**

Spec

```
CONTEXT GCy_ctx
EXTENDS Landing_Ctx
SETS
  GCylinders_Positions
CONSTANTS
  locked_up, unlocked_up, moving_down, locked_down
  pressure_high
AXIOMS
  axm_1: partition(GCylinders_Positions, {locked_up},
         {locked_down}, {moving_down}, {unlocked_up})
  axm_press_high: pressure_high ∈ ℕ
END
```

```
MACHINE GCy_mch
REFINES Landing_System
SEES GCy_ctx
VARIABLES
  op_mode, doors_pos, gears_pos //old variables
  gears_cyln // gears cylinders
  pressure // pressure on the hydraulic circuit
INVARIANTS
  inv_Typ_gears_cyln: gears_cyln ∈ GCylinders_Positions
  inv_Typ_pressure: pressure ∈ ℕ
  inv_order_1: gears_cyln = unlocked_up
                          ⇒ pressure = pressure_high
  glu_inv_1: ...
  glu_inv_2: ...
```

```
EVENTS
 INITIALISATION
   then
     ...
     act4: gears_cyln := locked_up
     act5: pressure := 0
   end
Event unlock_up_gears_cylinders
   when
      grd1: doors_pos = open
      grd2: gears_pos = g_retracted
      grd3: gears_cyln = locked_up
      grd_press: pressure = pressure_high
   then
      act1: gears_cyln := unlocked_up
   end
 Event move_down_gears_cylinders
   when
      grd1: doors_pos = open
      grd2: gears_cyln = unlocked_up
   then
      act1: gears_cyln := moving_down
   end
Event lock_down_gears_cylinders
  refines extend_gears
  when
     grd1: doors_pos = open
     grd2: gears_pos = g_retracted
     grd3: gears_cyln = moving_down
   then
     act1: gears_pos := g_extended
     act2: gears_cyln := locked_down
   end
 Event open_doors
...
```

GCy' est un enfant de LS et remplace LS-2 via le renommage.
Qu'est ce qui reste à compléter ? Il s'agit des deux invariants appelés *glu_inv_1* et *glu_inv_2*.

Le collage est introduit dans la machine GCy_mch par le spécifieur à travers les deux invariants de collage suivants :

glu_inv_1: gears_cyln = unlocked_up ⇒ doors_pos = open
glu_inv_2: gears_pos = g_extended ⇒ gears_cyln = locked_down

A.4.3 Prise en compte de DCy, frère de GCy

Nous partons du développement effectué en A.4.2.

Notre compréhension des besoins indique que le besoin DCy concerne les **détails** de la séquence d'ouverture des portes décrite également par le besoin LS-1.

CdC	
ID	*Description*
FUN-G	The [Landing_System] goal is maneuvering [gears_pos] and their associated [doors_pos]
...	...
LS	In [nominal] [op_mode], the landing sequence is:
LS-1	**1. [open_doors]**
GCy' // LS-2	2. When [gears_cyln] is [locked_up], and when it receives [pressure_high], [extend_gears] is:
GCy'-1	2.1. [unlock_up_gears_cylinders]
GCy'-2	2.2. [move_down_gears_cylinders]
GCy'-3	2.3. [lock_down_gears_cylinders]
LS-3	3. [close_doors]
DCy	When a door cylinder is locked in closed position and when it receives pressure from the extension hydraulic circuit, the opening doors sequence is:
DCy-1	1. unlock the door cylinder from the closed position
DCy-2	2. move it to the open position
DCy-3	3. maintain it in the open position as long as the pressure is maintained in the hydraulic extension circuit

est détaillé par

LS	In [nominal] [op_mode], the landing sequence is:
DCy // LS-1	**1. When a door cylinder is locked in closed position and when it receives pressure from the extension hydraulic circuit, the opening doors sequence is:**
DCy-1	**1. unlock the door cylinder from the closed position**
DCy-2	**2. move it to the open position**
DCy-3	**3. maintain it in the open position as long as the pressure is maintained in the hydraulic extension circuit**
GCy' // LS-2	2. When [gears_cyln] is [locked_up], and when it receives [pressure_high], [extend_gears] is:
GCy'-1	2.1. [unlock_up_gears_cylinders]
GCy'-2	2.2. [move_down_gears_cylinders]
GCy'-3	2.3. [lock_down_gears_cylinders]
LS-3	3. [close_doors]
...	...

- Dans la partie gauche de la figure ci-dessus, LS et DCy sont de même niveau.
- La partie décrite entre les deux lignes pointillées effectue un zoom sur le besoin LS-1.
- Dans la partie droite de la figure, nous montrons que DCy remplace LS-1 en devenant fils de LS.

Nous appliquons le patron Dev-seq pour DCy. Le système généré automatiquement est le suivant :

CdC

ID	*Description*
FUN-G	The [Landing_System] goal is maneuvering [gears_pos] and their associated [doors_pos]
...	...
LS	In [nominal] [op_mode], the landing sequence is:
DCy' // LS-1	**1. When [doors_cyln] is [locked_closed] and when it receives [pressure_extension], [open_doors] is:**
DCy'-1	**1.1. [unlock_closed_doors_cylinders]**
DCy'-2	**1.2. [move_open_doors_cylinders]**
DCy'-3	**1.3. [maintain_open_doors_cylinders]**
GCy' // LS-2	2. When [gears_cyln] is [locked_up], and when it receives [pressure_high], [extend_gears] is:
GCy'-1	2.1. [unlock_up_gears_cylinders]
GCy'-2	2.2. [move_down_gears_cylinders]
GCy'-3	2.3. [lock_down_gears_cylinders]
LS-3	3. [close_doors]

Glossaire

Formal term	*Informal description*
Landing_System	landing system
nominal	nominal
op_mode	mode, operating modes
emergency	emergency
gears_pos	gears, landing gears
doors_pos	doors
extend_gears	extending, extend the gears, extend the landing gears, the extension gears sequence
open_doors	open the doors, opening, **the opening doors sequence**
...	...
pressure_high	pressure from high hydraulic circuit
unlock_up_gears_cylinders	unlock the gear cylinder from the high position
move_down_gears_cylinders	move it to the down position
lock_down_gears_cylinders	lock it in the down position
doors_cyln	**a door cylinder**
locked_closed	**locked in closed position**
pressure_extension	**pressure from the extension hydraulic circuit**
unlock_closed_doors_cylinders	**unlock the door cylinder from the closed position**
move_open_doors_cylinders	**move it to the open position**
maintain_open_doors_cylinders	**maintain it in the open position as long as the pressure is maintained in the hydraulic extension circuit**

Spec

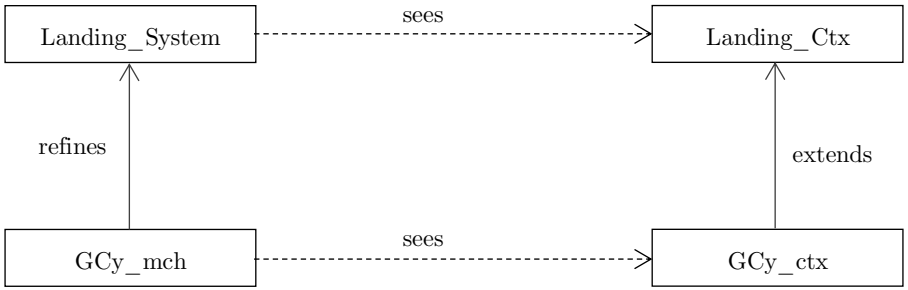
CONTEXT GCy_ctx
EXTENDS Landing_Ctx
SETS
  GCylinders_Positions, **DCylinders_Positions**
CONSTANTS
  locked_up, unlocked_up, moving_down, locked_down
  pressure_high, **pressure_extension**
  locked_closed, moving_open, maintained_open,
  unlocked_closed
AXIOMS
  axm_1: partition(GCylinders_Positions, {locked_up},
    {locked_down}, {moving_down}, {unlocked_up})
  axm_press_high: pressure_high ∈ ℕ
  axm_2: partition (DCylinders_Positions,
    {locked_closed}, {moving_open}, {maintained_open}
    {unlocked_closed})
  axm_press_ext: pressure_extension ∈ ℕ
END

MACHINE GCy mch
REFINES Landing_System
SEES GCy_ctx
VARIABLES
  op_mode, doors_pos, gears_pos //old variables
  gears cyln // gears cylinders
  pressure // pressure on the hydraulic circuit
  doors_cyln // doors cylinders
  pressure_doors
INVARIANTS
  inv_Typ_gears_cyln: gears_cyln ∈ GCylinders_Positions
  inv_Typ_pressure: pressure ∈ ℕ

```
 inv_order_1: gears_cyln = unlocked_up
                                  ⇒ pressure = pressure_high
 glu_inv1: gears_cyln = unlocked_up ⇒ doors_pos = open
 glu_inv2: gears_pos = g_extended
                        ⇒ gears_cyln = locked_down
 inv_Typ_doors_cyln: doors_cyln ∈ DCylinders_Positions
 inv_Typ_pressure_doors: pressure_doors ∈ ℕ
 inv_order_1: gears_cyln = unlocked_up
                        ⇒ doors_cyln = maintained_open
EVENTS
 INITIALISATION
  then
   ...
   act4: gears_cyln := locked_up
   act5: pressure := 0
   act6: doors_cyln := locked_closed
   act7: pressure_doors := 0
  end
 Event unlock_closed_doors_cylinders
 Event move_open_doors_cylinders
 Event maintain_open_doors_cylinders
  refines open_doors
  ...
 then
  act1: doors_pos := open
  act2: doors_cyln := maintained_open
 end
 Event unlock_up_gears_cylinders
  when
   grd1: doors_pos = open
   grd2: gears_pos = g_retracted
   grd3: gears_cyln = locked_up
   grd_press: pressure = pressure_high
   grd4: doors_cyln := maintained_open // order
  then
   act1: gears_cyln := unlocked_up
  end
 Event move_down_gears_cylinders
 Event lock_down_gears_cylinders
...
```

A.4.4 Utilisation des outils

A.4.4.1 Schéma des Specs développées

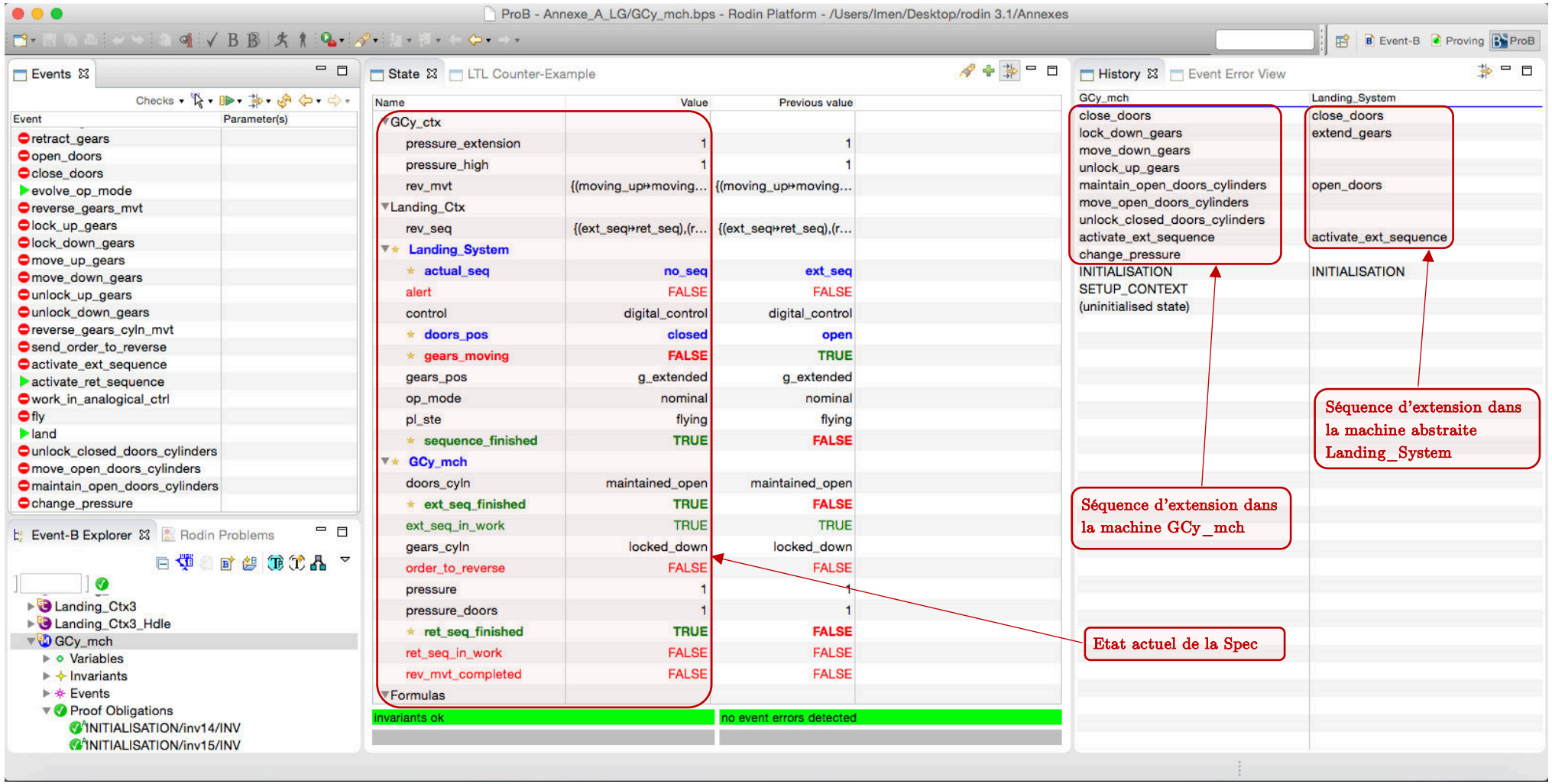


A.4.4.2 Vérification

Spec	***OPs automatiques***	***OPs interactives***	***Total***
Landing_Ctx	0	0	0
Landing_System	29	0	29
GCy_ctx	4	0	4
GCy_mch	47	8	55

A.4.4.3 Validation

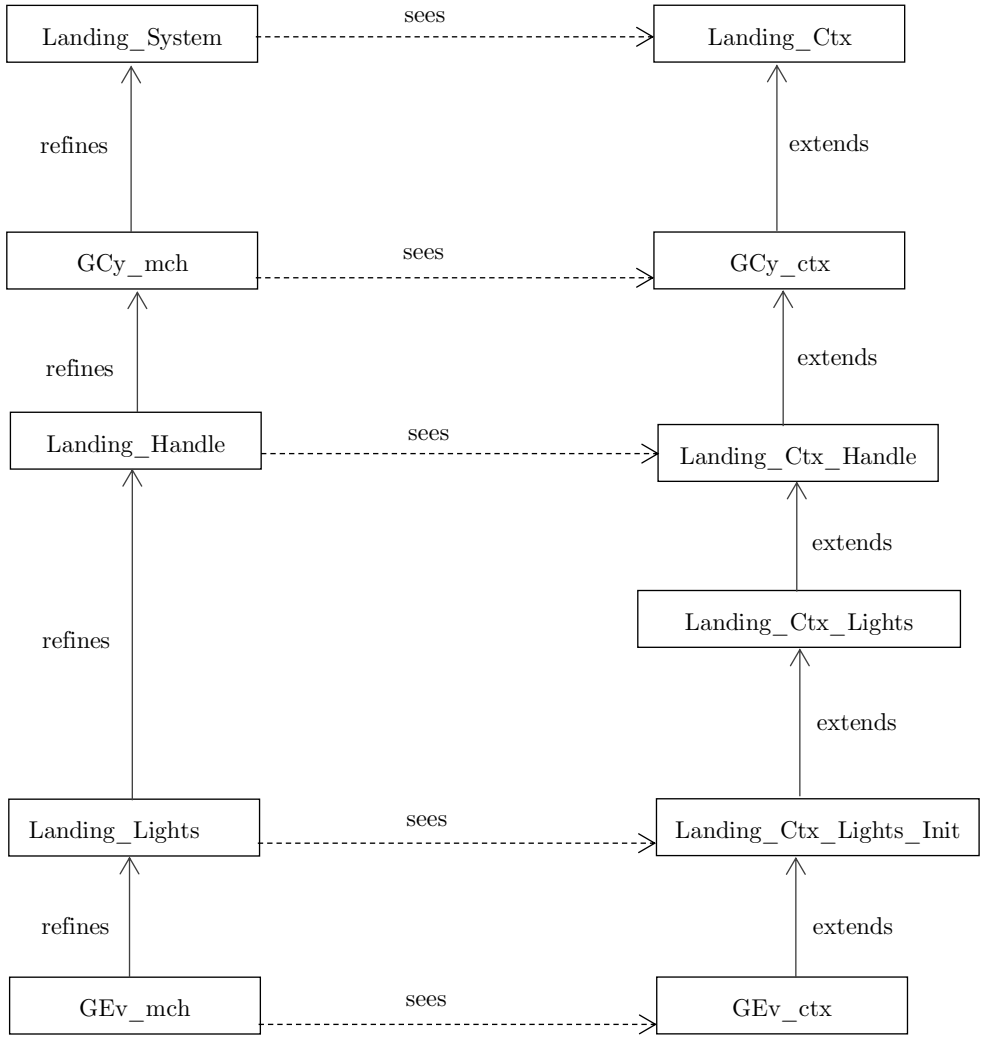
Animation de la séquence d'extension du train d'atterrissage à l'aide de ProB :



A.5 Evolution du développement

Nous avons travaillé sur les besoins *FUN-P-I* (et ses enfants) et *GEv* (voir CdC en A.2).

A.5.1 Schéma des Specs développées



Les Specs développées prennent en compte les besoins cités ci-dessus comme suit :

- *Landing_ Handle* et *Landing_ Ctx_ Handle* correspondent au développement des besoins *FUN-P-handle1* et *FUN-P-handle2*,
- *Landing_ Lights*, *Landing_ Ctx_ Lights* et *Landing_ Ctx_ Lights_ Init* représentent le développement associé aux besoins allant de *FUN-P-light1* à *FUN-P-light5* et
- *GEv_ mch* et *GEv_ ctx* correspondent au développement du besoin *GEv* en utilisant le patron Dev-seq. Ce développement n'est pas terminé.

A.5.2 Vérification

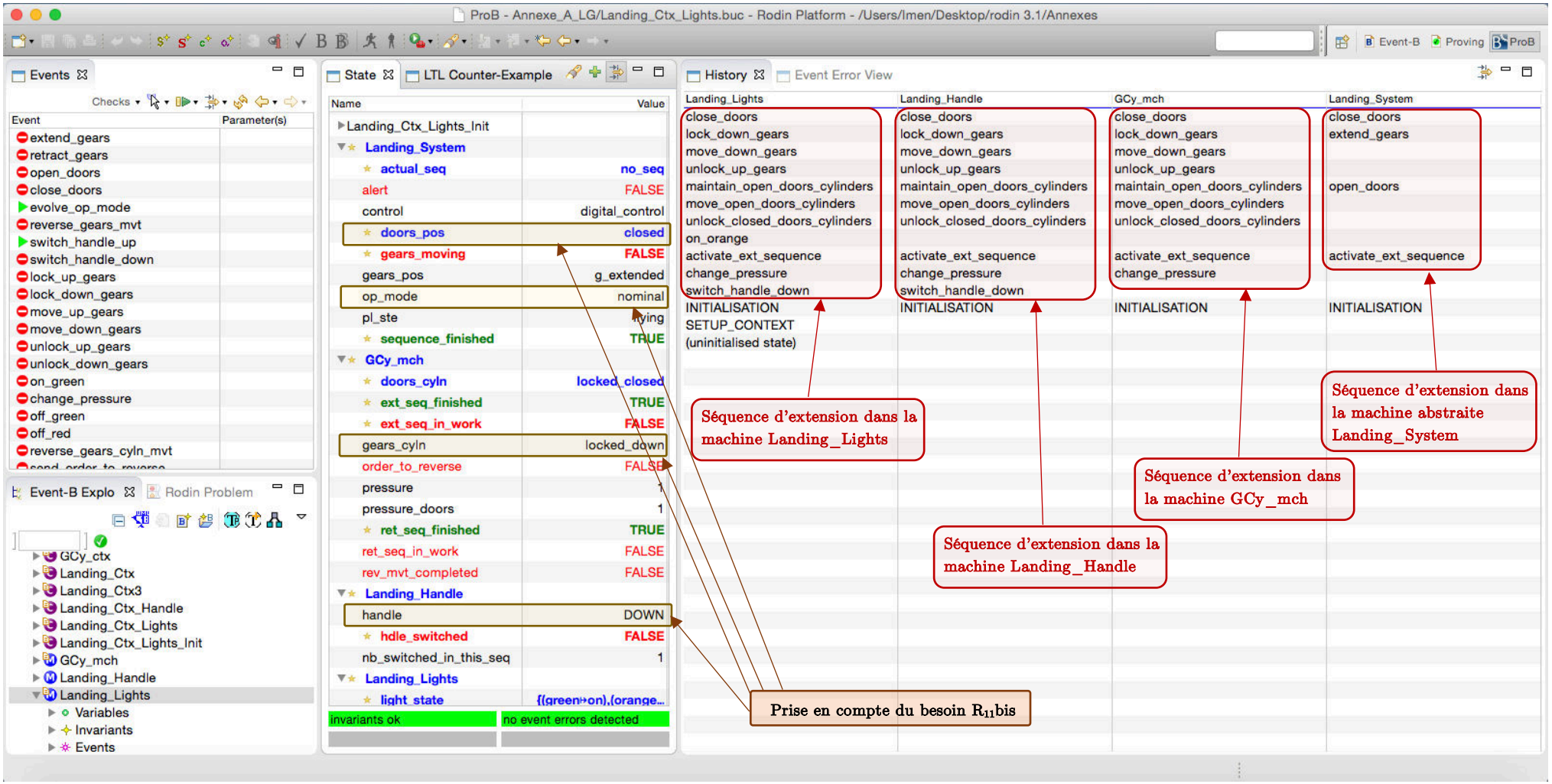
Spec	***OPs automatiques***	***OPs interactives***	***Total***
Landing_Ctx	0	0	0
Landing_System	29	0	29
GCy_ctx	4	0	4
GCy_mch	47	8	55
Landing_Ctx_Handle	0	0	0
Landing_Handle	56	12	68
Landing_Ctx_Lights	2	0	2
Landing_Ctx_Lights_Init	0	0	0
Landing_Lights	64	25	89
GEv_ctx	0	0	0
GEv_mch	83	11	94

A.5.3 Validation

Animation de la séquence d'extension du train par les Specs suivantes :

- La machine *Landing_ System* représente la Spec abstraite,
- *GCy_ mch* est son premier raffinement,
- *Landing_ Handle* est son deuxième raffinement et
- *Landing_ Lights* est son troisième raffinement.

L'animation de cette séquence dans la Spec *Landing_ Lights* est effectuée manuellement en déclenchant les événements concernés. Pour les autres Specs allantde *Landing_ System* à *Landing_ Handle*, cette animation est réalisée automatiquement à l'aide de l'outil ProB.



NB.

Le besoin $R_{11}bis$ décrit dans A.2 décrit à la fois une obligation et un comportement. Il est pris en compte lors de l'animation des machines *Landing_ Lights* et *Landing_ Handle*. Cette prise en compte se répercute par la valeur des quatre variables *op_ mode*, *handle*, *gears_ cyln* et *doors_ pos*.

A.6 Conclusion

Dans cette annexe, nous montrons quelques étapes de développement du système de train d'atterrissage d'un avion. Le système élaboré est constitué des besoins, leurs spécifications formelles associées et les liens entre eux mémorisés dans un glossaire. Nous partons du cahier des charges du client dans lequel nous nous occupons des besoins techniques décrits par des spécialistes du domaine et de ceux décrits dans la partie informatique (partie intitulée *« Requirements/Properties »*). Le développement a pris lieu grâce aux outils disponibles sous la plateforme Rodin. Nous avons effectué notre démarche selon deux manières différentes : la première consiste à un développement manuel des Specs et la deuxième se base sur un développement semi-automatique par instanciation de notre patron Dev-seq. Au fur et à mesure de la construction du système, la Spec est vérifiée et validée, le glossaire est mis à jour automatiquement et les besoins évoluent par introduction des termes formels dans leurs énoncés.