Annexe B
Machine d'hémodialyse

Le cahier des charges de cette étude de cas a été proposé dans la conférence ABZ 2016 [Mashkoor, 2016]. Nous nous occupons de la partie nommée « *4.2 Software requirements* » de ce document. Cette partie est décrite par des besoins ayant une forme identique. Notre objectif est de traiter les différents cas d'application du patron Dev-if sur ces besoins. Le développement réalisé est constitué des besoins, leurs spécifications formelles en B événementiel et les liens entre eux représentés par un glossaire. Ce développement est effectué sous la plateforme Rodin (version 3.2.0, http://rodin-b-sharp.sourceforge.net/) avec les plugins ProR, ProB et Project Diagram.

B.1 Quelques exigences du cahier des charges du client

R-5 *During initiation, if the software detects that the pressure at the VP transducer exceeds the upper pressure limit, then the software shall stop the BP and execute an alarm signal.*

R-6 *During initiation, if the software detects that the pressure at the VP transducer falls below the lower pressure limit, then the software shall stop the BP and execute an alarm signal.*

R-8 *During initiation, if the software detects that the pressure at the AP transducer falls below the lower pressure limit, then the software shall stop the BP and execute an alarm signal.*

R-10 *While connecting the patient, if the software detects that the pressure at the VP transducer falls below the defined lower pressure limit for more than 3 seconds, then the software shall stop the BP and execute an alarm signal.*

R-21 *If the machine is in the initiation phase and if the temperature falls below the minimum temperature of 33C, then the software shall disconnect the dialyzer from the DF and execute an alarm signal.*

R-14 *While connecting the patient, the software shall monitor the blood flow in the EBC and if no flow is detected, then the software shall stop the BP and execute an alarm signal. The blood flow can be detected by measuring the pump rotations with the speed sensor of the BP.*

R-15 *While connecting the patient, the software shall monitor the filling blood volume of the EBC and if the filling blood volume exceeds 400 mL, then the software shall stop the BP and execute an alarm signal. The blood volume can be detected by measuring the pump rotations with the speed sensor of the BP.*

R-16 *While connecting the patient, the software shall use a timeout of 310 seconds after the first start of the BP. After this timeout, the software shall change to the initiation phase.*

R-17 *While connecting the patient, the software shall monitor the blood flow direction and if the reverse direction is detected, then the software shall stop the BP and execute an alarm signal. The blood flow direction can be detected by the direction sensor of the BP.*

Rappel : forme générique conditionnelle des besoins

S_{p1} *if condition*(S_{p2}) *then action*(S_{p3})

où Sp_1 représente l'environnement du besoin en question.

B.2 Développement guidé par l'instanciation du patron Dev-if

Notre patron traite un seul besoin à la fois. Nous montrons la partie générée automatiquement et la partie à compléter désignée par « ... ».

B.2.1 Initialisation du développement

B.2.1.1 Prise en compte du besoin R-5

L'application de Dev-if sur R-5 génère automatiquement le résultat suivant :

CdC

ID	*Description*	*Event-B Model*
R-5'	[initiat] if [vp] exceeds [upper_press_limit] then stop [BP] and execute [alarm_vp]	**R-5'_Ctx, R-5'_Mch**
R-5'-init	[vp] is 0 [BP] is [stopped]	**R-5'_Ctx, R-5'_Mch**
R-5'-evolve	[vp] can [change_vp]	**R-5'_Ctx, R-5'_Mch**

Glossaire

Formal term	*Informal description*
initiat	**During initiation**
vp	**the pressure at the VP transducer**
upper_press_limit	**the upper pressure limit**
BP	**the BP**
alarm_vp	**an alarm signal**
stopped	**stop**
change_vp	**change the pressure at the VP transducer**

Spec

CONTEXT **R-5'_Ctx**
SETS
PHASES, BP_State, ALARMS
CONSTANTS
initiat, upper_press_limit, stopped, NULL, ALM_excess_vp
AXIOMS
axm1: partition (ℤ, {upper_press_limit})
axm2: partition (PHASES, {initiat})
axm31: partition (BP_State, {stopped})
axm32: partition (ALARMS, {NULL}, {ALM_excess_vp})
END

MACHINE **R-5'_Mch**
SEES **R-5'_Ctx**
VARIABLES
phase, vp, BP, alarm_vp
INVARIANTS
R-5'-phase-Typ: phase ∈ PHASES
R-5'-vp-Typ: vp ∈ ℤ
R-5'-BP-Typ: BP ∈ BP_State
R-5'-alarm_vp-Typ: alarm_vp ∈ ALARMS
R-5'-normal: ¬(vp > upper_press_limit) ∧ phase = initiat ⇒ ...
R-5'-anomaly: ... ⇒ vp > upper_press_limit ∧ phase = initiat
EVENTS
INITIALISATION
begin
R-5'-init-phase: phase := initiat
R-5'-init-vp: vp := 0
R-5'-init-alarm_vp: alarm_vp := NULL
R-5'-init-BP: BP := stopped
end

Event **change_vp**
any
value
where
grd1: value ∈ ℤ
then
R-5'-act: vp := value
end
Event **treatment_R-5'**
when
grd1: phase = initiat
grd2: vp > upper_press_limit
then
R-5'-act1: BP := stopped ∧
alarm_vp :=ALM_excess_vp
end
END

Qu'est-ce qu'il reste à compléter ?

- Les deux invariants *R-5'-normal* et *R-5'-anomaly* introduits dans la machine *R-5'_Mch* sont complétés par le spécifieur comme suit :

R-5'-normal: $\neg(vp > upper\_press\_limit) \wedge phase = initiat \Rightarrow alarm\_vp = NULL$
R-5'-anomaly: $alarm\_vp = ALM\_excess\_vp \Rightarrow vp > upper\_press\_limit \wedge phase = initiat$

- Vu qu'il s'agit d'une première application du patron Dev-seq, nous avons besoin d'exprimer la dynamique de la Spec en introduisant :
  - de nouvelles constantes « *preparation* », « *connect_patient* » et « *therapy_ending* » modélisant trois phases de la thérapie,
  - deux événements pour démarrer et arrêter la pompe à sang BP,
  - des événements pour gérer les changements de phases de l'hémodialyse et
  - un événement pour réinitialiser la Spec après détection d'une anomalie.

Ces événements sont décrits comme suit :

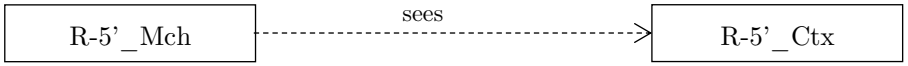
```
Event start_blood_pumping  //démarrer la pompe à sang
    where
      grd1: BP = stopped
      grd2: alarm_vp = NULL
    then
      act1: BP := started
    end
Event stop_blood_pumping  // arrêter la pompe à sang
    when
      grd1: BP = started
    then
      act1: BP := stopped
      act2: vp := 0
    end
```

```
Event change_preparation_to_initiation
    when
      grd1: phase = preparation
      grd2: BP = stopped
      grd3: alarm_vp = NULL
    then
      act1: phase := initiat
    end
Event change_initiation_to_therapy_ending
    when
      grd1: phase = initiat
      grd2: BP = stopped
      grd3: alarm_vp = NULL
    then
      act1: phase := therapy_ending
    end
... // et autres événements
```

```
Event reinitialise // réintialiser l'état du système
    where
      grd1: ∃ a · (a ∈ ALARMS ∧ a ≠ NULL) // au moins une alarme déclenchée
    then
      act1: phase := initiat
      act2: vp := 0
      act3: ∀ a . a ∈ ALARMS | a := NULL // remettre toutes les alarmes à NULL
    end
```

B.2.1.2 Utilisation des outils

a) Schéma des Specs développées (outil : Project Diagram)



b) Vérification (outils : générateurs d'OPs et prouveurs)

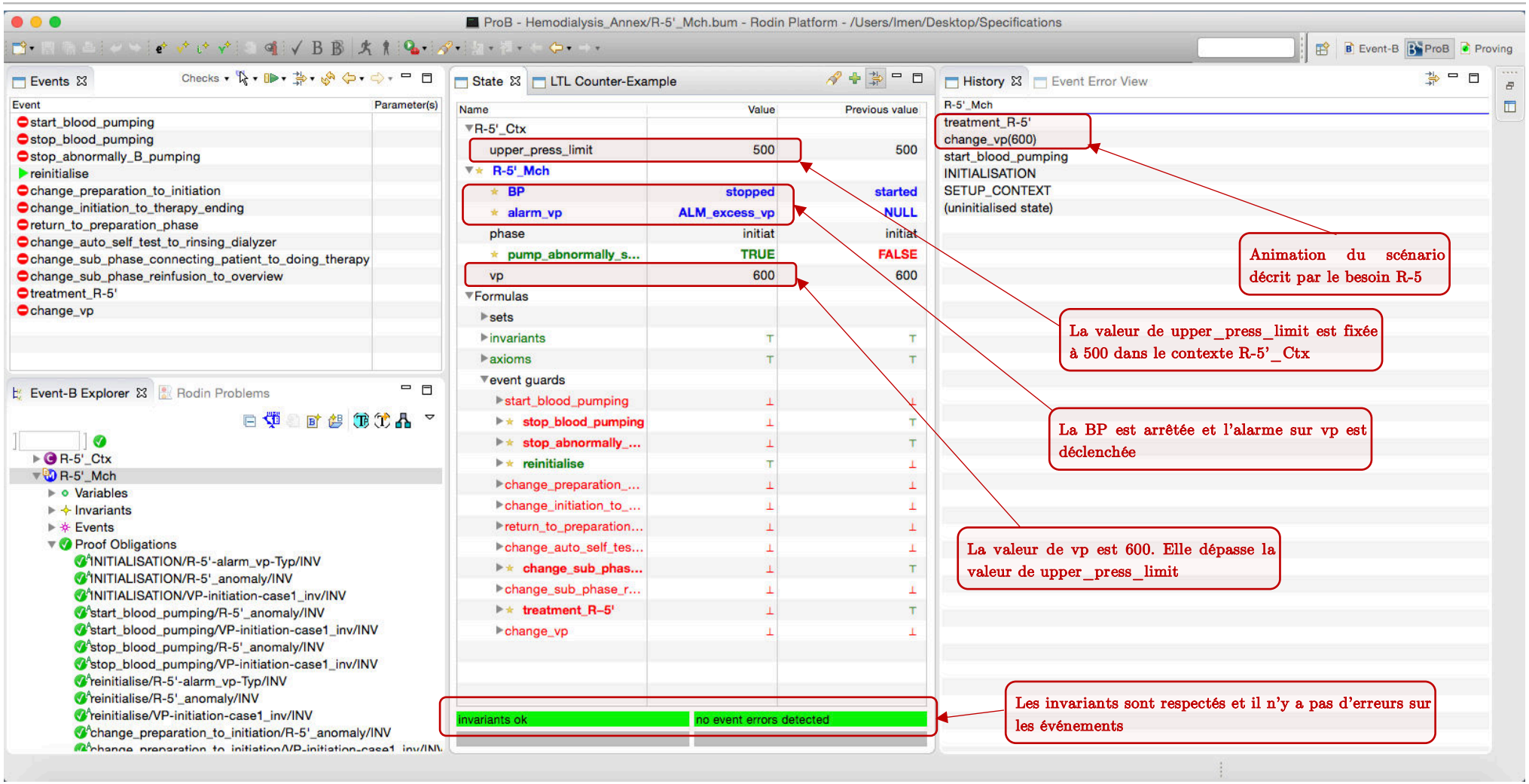
Spec	***OPs automatiques***	***OPs interactives***	***Total***
R-5'_ Mch	21	0	21

c) Validation (outil : ProB)

Nous traitons le scénario décrit par le besoin R-5 et présenté comme suit :

$$\left.\begin{array}{l} \textit{Initiation phase} \\ \quad \wedge \\ \textit{vp exceeds the upper pressure} \\ \textit{limit} \end{array}\right\} \Rightarrow \left\{\begin{array}{l} \textit{stop BP} \\ \quad \wedge \\ \textit{execute an alarm signal} \end{array}\right.$$

Le scénario présenté ci-dessus est décrit par l'animation de la machine en suivant l'enchaînement des événements *change_vp(600)* $\rightarrow$ *treatment_R-5'* (voir la capture d'écran prise lors de l'utilisation de l'outil ProB sous Rodin).



La machine *R-5'_Mch* est valide relativement au scénario décrit dans le besoin R-5.

B.2.2 Prise en compte de la condition

Afin de se situer dans ce cas, nous effectuons une comparaison entre deux besoins, par exemple entre R-5 et R-6.

B.2.2.1 Comparaison entre R-5 et R-6

Ces deux besoins ont :

- les mêmes paramètres :
  - $\mathbf{S_{p1}}$: « *initiation* »
  - $\mathbf{S_{p2}}$: « *pressure at the VP transducer* »
  - $\mathbf{S_{p3}}$: « *BP* » et « *an alarm signal* »
- les mêmes actions(S_{p3}) :
  - stop BP
  - execute an alarm signal
- une clause *if* différente :
  - dans **R-5** : « *the software detects that the pressure at the VP transducer exceeds the upper pressure limit* »
  - dans **R-6** : « *the software detects that the pressure at the VP transducer falls below the lower pressure limit* »

B.2.2.2 Application de Dev-if sur R-6

Le patron utilise le développement existant de R-5 (son CdC, sa Spec et son glossaire) effectué en **B.2.1**. Le système résultant est le suivant :

CdC

ID	*Description*	*Event-B Model*
R-5'	[initiat] if [vp] exceeds [upper_press_limit] then stop [BP] and execute [alarm_vp]	R-5'_Ctx, R-5'_Mch
R-5'-init	[vp] is 0 [BP] is [stopped]	R-5'_Ctx, R-5'_Mch
R-5'-evolve	[vp] can [change_vp]	R-5'_Ctx, R-5'_Mch
R-6'	**[initiat] if [vp] falls below [lower_press_limit] then stop [BP] and execute [alarm_vp]**	**R-6'_Ctx, R-6'_Mch**
R-6'-evolve	**[vp] can [change_vp]**	**R-6'_Ctx, R-6'_Mch**

Glossaire

Formal term	*Informal description*
initiat	During initiation
vp	the pressure at the VP transducer
upper_press_limit	the upper pressure limit
BP	the BP
alarm_vp	an alarm signal
stopped	stop
change_vp	change the pressure at the VP transducer
lower_press_limit	**the lower pressure limit**

Spec

CONTEXT **R-6'_Ctx**
EXTENDS **R-5'_Ctx**
SETS
CONSTANTS
lower_press_limit, ALM_deficit_vp
AXIOMS
axm1: partition (ℤ, {lower_press_limit})
axm32: partition (ALARMS, {NULL}, {ALM_deficit_vp})
END

MACHINE **R-6'_Mch**
REFINES **R-5'_Mch**
SEES **R-6'_Ctx**
VARIABLES
phase, vp, BP, alarm_vp
INVARIANTS
R-6'-normal: ¬(vp < lower_press_limit) ∧ phase = initiat ⇒ ...
R-6'-anomaly: ... ⇒ vp < lower_press_limit ∧ phase = initiat
EVENTS
INITIALISATION // mêmes initialisations
Event **change_vp // provenant de R-5'_Mch // reste le même pour R-6**
any
value
where
grd1: value ∈ ℤ
then
R-5'-act: vp := value
end

Event **treatment_R-6'**
when
grd1: phase = initiat
grd2: vp < lower_press_limit
then
R-6'-act1: BP := stopped ∧ alarm_vp :=ALM_deficit_vp
end
Event treatment_R-5' // hérité de R-5'_Mch
when
grd1: phase = initiat
grd2: vp > upper_press_limit
then
R-5'-act1: BP := stopped ∧ alarm_vp :=ALM_excess_vp
end
...
END

Ce qui reste à compléter est :

R-6'-normal: $\neg(vp < lower\_press\_limit) \wedge phase = initiat \Rightarrow$ **alarm_vp = NULL**
R-6'-anomaly: **alarm_vp = ALM_deficit_vp** $\Rightarrow vp < lower\_press\_limit \wedge phase = initiat$

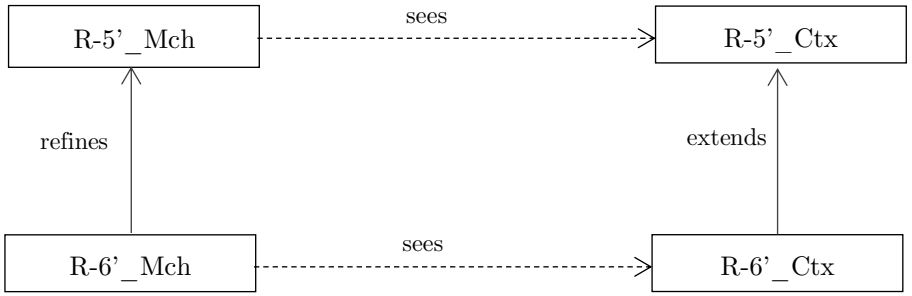
Nous enrichissons l'invariant *R-6'-normal* en introduisant une propriété provenant de R-5 comme suit :

R-6'-normal: $\neg(vp < lower\_press\_limit) \wedge phase = initiat \wedge \neg(vp > upper\_press\_limit) \Rightarrow$ **alarm_vp = NULL**

Cet invariant assure que lorsque la pression sur vp est dans l'intervalle **[lower_press_limit, upper_press_limit]**, l'alarme n'est pas déclenchée.

B.2.2.3 Utilisation des outils

a) Schéma des Specs développées



b) Vérification

Spec	*OPs automatiques*	*OPs interactives*	*Total*
R-5'_ Mch	21	0	21
R-6'_Mch	19	4	23

Les 4 OPs interactives de R-6'_Mch concernent une propriété spécifique appelée *« EQL »*. Celle-ci permet de s'assurer que les valeurs des variables abstraites ne sont pas changées dans les raffinements. Ceci n'est pas le cas lors de l'application de Dev-if sur R-6 en utilisant le développement existant de R-5. Les

variables *BP* et *alarm_vp* sont abstraites vu qu'elles proviennent de la machine *R-5'_Mch*. Dans la machine raffinée *R-6'_Mch*, les valeurs de ces variables sont changées (à travers l'événement *treatment_R-6')* afin de prendre en compte le besoin R-6. Nous avons prouvé manuellement ces 4 OPs en utilisant le contre-prouveur de ProB.

c) Validation

Elle est prise en compte au fur et à mesure de l'évolution du développement du système. Nous effectuons l'animation de *R-6'_Mch*. Cette machine est valide relativement au besoin R-6. De plus, le champ *Event-B Model* du CdC indique que le besoin *R-6'* et son enfant *R-6'-evolve* sont pris en compte par la machine *R-6'_Mch* et son contexte *R-6'_Ctx*.

B.2.3 Prise en compte des paramètres de la condition

B.2.3.1 Comparaison entre R-6 et R-8

Ces deux besoins ont :

- la même clause *if* : *« the software detects that the pressure at the VP transducer falls below the lower pressure limit »*
- les mêmes paramètres :
  - $\mathbf{S_{p1}}$ **:** « *initiation*
  - $\mathbf{S_{p3}}$ **:** *« BP »* et *« an alarm signal »*
- les mêmes actions mêmes actions (S_{p3}) :
  - stop BP
  - execute an alarm signal
- un paramètre de $\mathbf{S_{p2}}$ différent :
  - dans **R-6 :** *« the pressure at the **VP** transducer »*
  - dans **R-8 :** *« the pressure at the **AP** transducer »*

B.2.3.2 Application de Dev-if sur R-8

Le patron utilise le développement existant de R-6 effectué en **B.2.2.2**.

a) Résultat automatique

CdC

ID	*Description*	*Event-B Model*
R-5'	[initiat] if [vp] exceeds [upper_press_limit] then stop [BP] and execute [alarm_vp]	R-5'_Ctx, R-5'_Mch
R-5'-init	[vp] is 0 [BP] is [stopped]	R-5'_Ctx, R-5'_Mch
R-5'-evolve	[vp] can [change_vp]	R-5'_Ctx, R-5'_Mch
R-6'	[initiat] if [vp] falls below [lower_press_limit] then stop [BP] and execute [alarm_vp]	R-6'_Ctx, R-6'_Mch
R-6'-evolve	[vp] can [change_vp]	R-6'_Ctx, R-6'_Mch
R-8'	**[initiat] if [ap] falls below [lower_press_limit] then stop [BP] and execute [alarm_ap]**	**R-8'_Ctx, R-8'_Mch**
R-8'-init	**[ap] is 0**	**R-8'_Ctx, R-8'_Mch**
R-8'-evolve	**[ap] can [change_ap]**	**R-8'_Ctx, R-8'_Mch**
glue-R-8'-R-6	**...**	**R-8'_Ctx, R-8'_Mch**

Glossaire

Formal term	*Informal description*
initiat	During initiation
vp	the pressure at the VP transducer
upper_press_limit	the upper pressure limit
BP	the BP
alarm_vp	an alarm signal
stopped	stop
change_vp	change the pressure at the VP transducer
lower_press_limit	the lower pressure limit
ap	**the pressure at the AP transducer**
alarm_ap	**an alarm signal**
change_ap	**change the pressure at the AP transducer**

Spec

CONTEXT **R-8'_Ctx**
EXTENDS **R-6'_Ctx**
SETS
CONSTANTS
ALM_deficit_ap
AXIOMS
axm32: partition (ALARMS, {ALM_deficit_ap})
END

MACHINE **R-8'_Mch**
REFINES **R-6'_Mch**
SEES **R-8'_Ctx**
VARIABLES
phase, vp, BP, alarm_vp
ap, alarm_ap
INVARIANTS
R-8'-ap-Typ: ap ∈ ℤ
R-8'-alarm_ap-Typ: alarm_ap ∈ ALARMS
R-8'-normal: ¬(vp < lower_press_limit) ∧ phase = initiat ⇒ ...
R-8'-anomaly: ... ⇒ vp < lower_press_limit ∧ phase = initiat
glue-R-8'-R-6: ... ⇒ ap < lower_press_limit ∧ phase = initiat ∧ ...
EVENTS
INITIALISATION
...
R-5'-init-BP: BP := stopped
R-8'-init-ap: ap := 0
R-8'-init-alarm_vp: alarm_vp := NULL
end
Event **change_ap**
any
value
where
grd1: value ∈ ℤ
then
R-8'-act: ap := value
end

Event **treatment_R-8'**
when
grd1: phase = initiat
grd2: ap < lower_press_limit
then
R-8'-act1: BP := stopped ∧ alarm_ap :=ALM_deficit_ap
end
Event **glue_treatment_R-8'**
when
grd1: ... // condition in R-6
grd2: phase = initiat
grd3: ap < lower_press_limit
then
R-8'-act1: BP := stopped ∧ alarm_ap := ALM_deficit_ap
R-8'-act2: ... // action in R-6
end
Event change_vp // hérité de R-6'_Mch
Event treatment_R-6' // hérité de R-6'_Mch
Event treatment_R-5' // hérité de R-5'_Mch
...
END

Il reste à compléter les « ... ».

b) Eléments complétés par le spécifieur

- Dans le ***CdC*** : expression du collage entre R-6 et R-8

ID	*Description*	*Event-B Model*
glue-R-8'-R-6	**[initiat] if [ap] and [vp] fall below [lower_press_limit] then stop [BP] and execute [alarm_ap] and [alarm_vp]**	R-8'_Ctx, R-8'_Mch

- Dans la ***Spec*** :
  - Invariants

R-8'-normal: ¬(vp < lower_press_limit) ∧ phase = initiat ⇒ **alarm_ap = NULL**
R-8'-anomaly: **alarm_ap = ALM_deficit_ap** ⇒ vp < lower_press_limit ∧ phase = initiat
glue-R-8'-R-6: **alarm_ap = ALM_deficit_ap ∧ alarm_vp = ALM_deficit_vp** ⇒ ap < lower_press_limit ∧ phase = initiat ∧ **vp < lower_press_limit** // invariant de collage

- Événement de collage entre R-6 et R-8 :

```
Event glue_treatment_R-8'
    when
      grd1: vp < lower_press_limit // condition in R-6
      grd2: phase = initiat
      grd3: ap < lower_press_limit
    then
      R-8'-act1: BP := stopped ∧ alarm_ap := ALM_deficit_ap
      R-8'-act2: alarm_vp := ALM_deficit_vp // action dans R-6
    end
```

- Événement de collage entre R-6 et R-8 pour atteindre l'état où les deux conditions(S_{p2}) sont vraies en même temps :

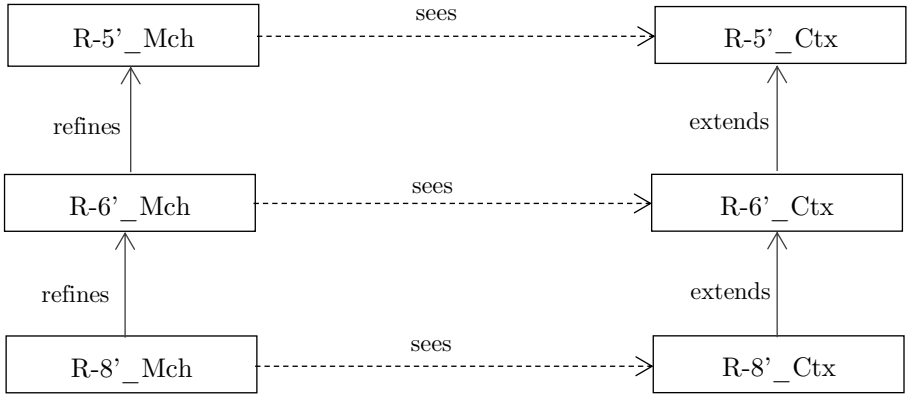
```
Event change_ap_vp
    refines change_vp // provenant de R-5' et de R-6'
    any
      value, value2
    where
      grd1: value ∈ ℤ
      grd2: value2 ∈ ℤ
    then
       R-5'-act: vp := value
      R-8'-act: ap := value2
    end
```

- Événement de collage entre R-5 et R-8 : cet événement découle de notre propre compréhension du système. Il prévoit le comportement commun dans le cas où les deux besoins R-5 et R-8 ont lieu en même temps :

```
Event glue_treatment_R-8'_R-5'
    when
      grd1: vp > upper_press_limit // condition dans R-5
      grd2: phase = initiat
      grd3: ap < lower_press_limit
    then
      R-8'-act1: BP := stopped ∧ alarm_ap := ALM_deficit_ap
      R-8'-act2: alarm_vp := ALM_excess_vp // action dans R-5
    end
```

B.2.3.3 Utilisation des outils

a) Schéma des Specs développées

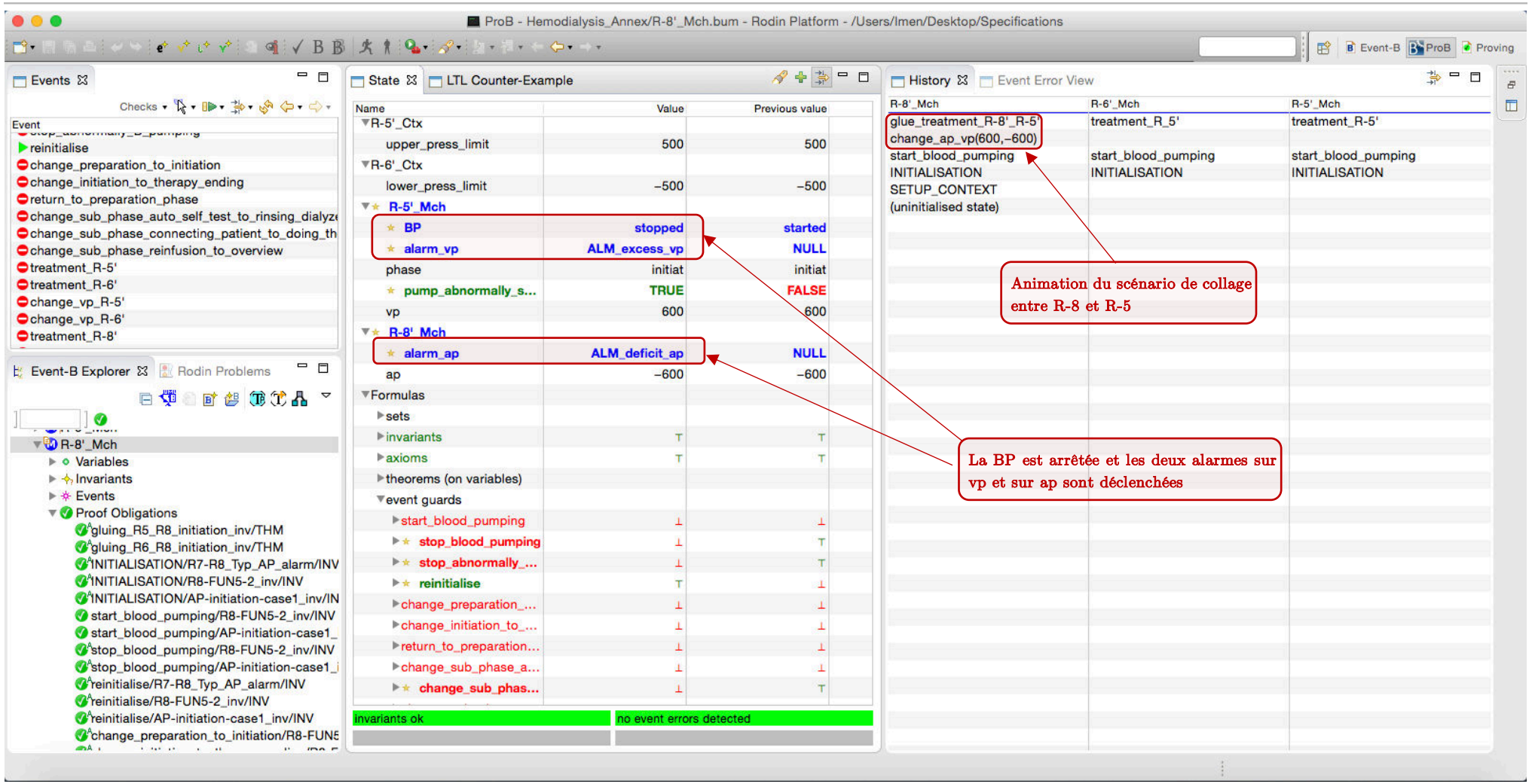


b) Vérification

Spec	*OPs automatiques*	*OPs interactives*	*Total*
R-5'_ Mch	21	0	21
R-6'_Mch	19	4	23
R-8'_Mch	36	2	38

c) Validation

Nous exécutons un scénario de **collage** entre **R-8** et **R-5** à l'aide de ProB, voir la capture d'écran de la page suivante.


ProB - Hemodialysis_Annex/R-8'_Mch.bum - Rodin Platform - /Users/lmen/Desktop/Specifications
Events
reinitialise
change_preparation_to_initiation
change_initiation_to_therapy_ending
return_to_preparation_phase
change_sub_phase_reinfusion_to_overview
treatment_R-5'
treatment_R-6'
change_vp_R-5'
change_vp_R-6'
treatment_R-8'
Event-B Explorer
Rodin Problems
R-8'_Mch
Variables
Invariants
Events
Proof Obligations
State
LTL Counter-Example
R-5'_Ctx
upper_press_limit 500 500
R-6'_Ctx
lower_press_limit −500 −500
R-5'_Mch
BP stopped started
alarm_vp ALM_excess_vp NULL
phase initiat initiat
pump_abnormally_s... TRUE FALSE
vp 600 600
R-8'_Mch
alarm_ap ALM_deficit_ap NULL
ap −600 −600
Formulas
sets
invariants
axioms
theorems (on variables)
event guards
start_blood_pumping
stop_blood_pumping
stop_abnormally_...
reinitialise
change_preparation_...
change_initiation_to_...
return_to_preparation...
change_sub_phase_a...
change_sub_phas...
invariants ok
no event errors detected
History
Event Error View
R-8'_Mch
R-6'_Mch
R-5'_Mch
glue_treatment_R-8'_R-5'
change_ap_vp(600,−600)
start_blood_pumping
INITIALISATION
SETUP_CONTEXT
(uninitialised state)
treatment_R_5'
start_blood_pumping
INITIALISATION
treatment_R-5'
start_blood_pumping
INITIALISATION
Animation du scénario de collage entre R-8 et R-5
La BP est arrêtée et les deux alarmes sur vp et sur ap sont déclenchées

B.2.4 Prise en compte de l'environnement

Le patron utilise le développement existant de R-8 effectué en **B.2.3**.

B.2.4.1 Prise en compte de R-10 (version complétée par le spécifieur)

CdC

ID	*Description*	*Event-B Model*
...	...	...
R-8'	[initiat] if [ap] falls below [lower_press_limit] then stop [BP] and execute [alarm_ap]	R-8'_Ctx, R-8'_Mch
R-8'-init	[ap] is 0	R-8'_Ctx, R-8'_Mch
R-8'-evolve	[ap] can [change_ap]	R-8'_Ctx, R-8'_Mch
glue-R-8'-R-6	[initiat] if [ap] and [vp] fall below [lower_press_limit] then stop [BP] and execute [alarm_ap] and [alarm_vp]	R-8'_Ctx, R-8'_Mch
R-10'	**[connect_patient] if [vp] falls below [lower_press_limit] for [duration_vp] more than 3 seconds then stop [BP] and execute [alarm_vp]**	**R-10'_Ctx, R-10'_Mch**
R-10'-init	**[duration_vp] is 0**	**R-10'_Ctx, R-10'_Mch**
R-10'-evolve	**[duration_vp] can [change_duration_vp]**	**R-8'_Ctx, R-8'_Mch**

Glossaire

Formal term	*Informal description*
...	...
lower_press_limit	the lower pressure limit, **the defined lower pressure limit**
ap	the pressure at the AP transducer
alarm_ap	an alarm signal
change_ap	change the pressure at the AP transducer
connect_patient	**While connecting the patient**
duration_vp	**a duration**
change_duration_vp	**change a duration**

Spec

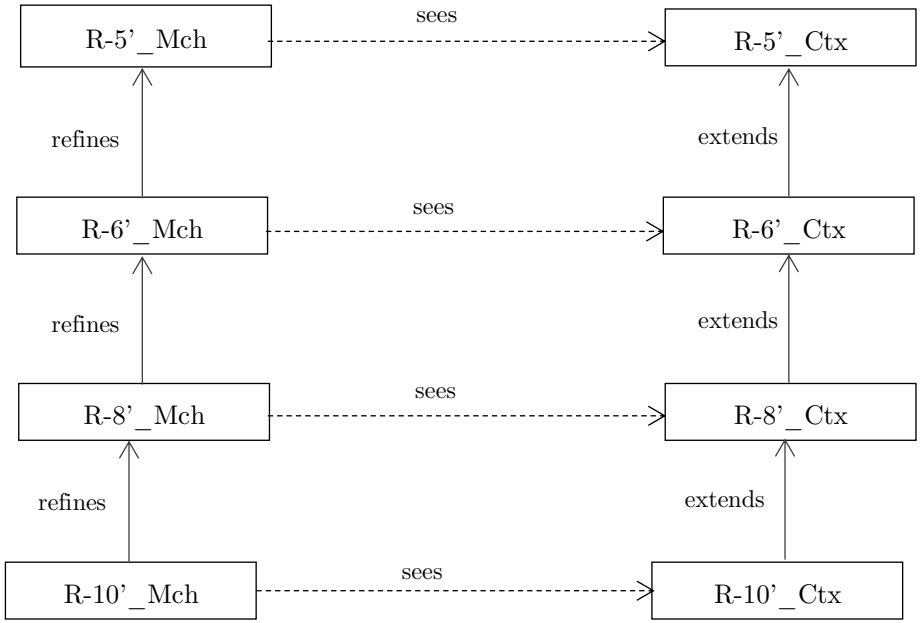
CONTEXT **R-10'_Ctx**
EXTENDS **R-8'_Ctx**
SETS
CONSTANTS
connect_patient, ALM_deficit_vp_connect
AXIOMS
axm2: partition (PHASES, {connect_patient})
axm3: partition (ALARMS, {ALM_deficit_vp_connect})
END

MACHINE **R-10'_Mch**
REFINES **R-8'_Mch**
SEES **R-10'_Ctx**
VARIABLES
phase, vp, BP, alarm_vp, ap, alarm_ap
duration_vp
INVARIANTS
R-10'-duration_vp-Typ : duration_vp ∈ ℕ
R-10'-normal: ¬(vp < lower_press_limit ∧ duration_vp > 3) ∧ phase = connect_patient ⇒ alarm_vp = NULL
R-10'-anomaly: alarm_vp = ALM_deficit_vp_connect ⇒ vp < lower_press_limit ∧duration_vp > 3 ∧ phase = connect_patient
EVENTS
INITIALISATION
begin
R-10'-init-phase: phase := connect_patient
R-10'-init-duration_vp: duration_vp := 0
end
Event **change_duration_vp**
any
value
where
grd1: value ∈ ℕ
then
R-10'-act: duration_vp := value
end

Event **treatment_R-10'**
when
grd1: phase = connect_patient
grd2: vp < lower_press_limit ∧ duration_vp > 3
then
R-10'-act1: BP := stopped ∧ alarm_vp :=ALM_deficit_vp_connect
end
Event change_ap // hérité de R-8'_Mch
Event treatment_R-8' // hérité de R-8'_Mch
Event glue_treatment_R-8' // hérité de R-8'_Mch
Event change_vp // hérité de R-6'_Mch
Event treatment_R-6' // hérité de R-6'_Mch
Event treatment_R-5' // hérité de R-5'_Mch
...
END

B.2.4.2 Utilisation des outils

a) Schéma des Specs développées



b) Vérification

Spec	***OPs automatiques***	***OPs interactives***	***Total***
R-5'_ Mch	21	0	21
R-6'_Mch	19	4	23
R-8'_Mch	36	2	38
R-10'_Mch	27	11	38

c) Validation

L'animation de la machine *R-10'_Mch* a été effectuée avec succès.

B.2.5 Cas général

Nous montrons deux cas d'application du Dev-if sur le besoin R-21 en utilisant :

- le développement existant de R-10 ou
- le développement existant de R-8.

Chaque cas dépend du degré de différence entre les deux besoins en question.

B.2.5.1 Prise en compte de R-21 relativement à R-10

Nous partons du développement existant de R-10 effectué en **B.2.4**.

a) Comparaison entre R-21 et R-10

	R-21	R-10
S_{p1}	If the machine is in the initiation phase	While connecting the patient
S_{p2}	the temperature	- the pressure at the VP transducer - a duration » (déduit à partir de notre compréhension des termes « for more than 3 seconds »)
S_{p3}	- the dialyser - an alarm signal	- the BP - an alarm signal
clause if	the temperature falls below the minimum temperature of 33C	the pressure at the VP transducer falls below the defined lower pressure limit for more than 3 seconds
clause then	the software shall disconnect the dialyser from the DF and execute an alarm signal	the software shall stop the BP and execute an alarm signal

Tous les paramètres sont différents entre les deux besoins.

b) Instanciation de Dev-if sur R-21

CdC

ID	*Description*	*Event-B Model*
...	...	...
glue-R-8′-R-6	[initiat] if [ap] and [vp] fall below [lower_press_limit] then stop [BP] and execute [alarm_ap] and [alarm_vp]	R-8'_Ctx, R-8'_Mch
R-10'	[connect_patient] if [vp] falls below [lower_press_limit] for [duration_vp] more than 3 seconds then stop [BP] and execute [alarm_vp]	R-10'_Ctx, R-10'_Mch
R-10′-init	[duration_vp] is 0	R-10'_Ctx, R-10'_Mch
R-10′-evolve	[duration_vp] can [change_duration_vp]	R-8'_Ctx, R-8'_Mch
R-21'	**[initiat] if [temperature] falls below 33C then disconnect [dialyser] and execute [alarm_temp]**	**R-21'_Ctx, R-21'_Mch**
R-21′-init	**[temperature] is 37 [dialyser] is [disconnected]**	**R-21'_Ctx, R-21'_Mch**
R-21′-evolve	**[temperature] can [change_temperature]**	**R-21'_Ctx, R-21'_Mch**

Glossaire

Formal term	*Informal description*
...	...
lower_press_limit	the lower pressure limit, the defined lower pressure limit
ap	the pressure at the AP transducer
alarm_ap	an alarm signal
change_ap	change the pressure at the AP transducer
connect_patient	While connecting the patient
duration_vp	a duration
change_duration_vp	change a duration
temperature	**the temperature**
dialyser	**the dialyser**
alarm_temp	**an alarm signal**
disconnected	**disconnected**
change_temperature	**change the temperature**

Spec

CONTEXT **R-21'_Ctx**
EXTENDS **R-10'_Ctx**
SETS
DIALYSER_STATE
CONSTANTS
connected, disconnected, ALM_deficit_temp
AXIOMS
axm31: partition (DIALYSER_STATE, {connected}, {disconnected})
axm32: partition (ALARMS, {ALM_deficit_temp})
END

MACHINE **R-21'_Mch**
REFINES **R-10'_Mch**
SEES **R-21'_Ctx**
VARIABLES
phase, vp, BP, alarm_vp, ap, alarm_ap, duration_vp
temperature, dialyser, alarm_temp
INVARIANTS
R-21'-temperature-Typ: temperature ∈ ℕ
R-21'-dialyser-Typ: dialyser ∈ DIALYSER_STATE
R-21'-alarm_temp-Typ: alarm_temp ∈ ALARMS
R-21'-normal: ¬(temperature < 33 ∧ phase = initiat) ⇒ alarm_temp = NULL
R-21'-anomaly: alarm_temp = ALM_deficit_temp ⇒ temperature < 33 ∧ phase = initiat
EVENTS
INITIALISATION
begin
R-21'-init-temperature: temperature := 37
R-21'-init-dialyser: dialyser := disconnected
End
Event **change_temperature**
any
value
where
grd1: value ∈ ℕ
then
R-21'-act: temperature := value
end

Event **treatment_R-21'**
when
grd1: phase = initiat
grd2: temperature < 33
then
R-21'-act1: dialyser := disconnected ∧ alarm_temp :=ALM_deficit_temp
end
Event **connect_dialyser //événement ajouté après instanciation du patron**
when
grd1: phase = initiat
grd2: dialyser = disconnected
grd3: temperature < 33
grd4: ∀ a . a ∈ ALARMS | a = NULL // aucune alarme n'est déclenchée
then
R-21'-act1: dialyser := connected
end
Event change_duration_vp // hérité de R-10'_Mch
Event treatment_R-10' // hérité de R-10'_Mch
Event change_ap // hérité de R-8'_Mch
Event treatment_R-8' // hérité de R-8'_Mch
Event glue_treatment_R-8′ // hérité de R-8'_Mch
Event change_vp // hérité de R-6'_Mch
Event treatment_R-6' // hérité de R-6'_Mch
Event treatment_R-5' // hérité de R-5'_Mch
...
END

NB.

1) Dans ce développement, il n'y a pas de collage entre les deux besoins R-21 et R-10 vu qu'ils sont situés dans deux phases différentes de thérapie :
- R-21 dans la phase d'initiation
- R-10 dans la phase de connexion du patient

2) Après révision de *R-21'_Mch*, nous remarquons qu'il est possible d'améliorer cette Spec en introduisant le collage avec le développement de R-8. Ceci est dû au fait que le développement de R-21 raffine celui de R-8 en passant par la Spec de R-10. Ces deux besoins, R-8 et R-21, se situent dans la même phase de thérapie, l'initiation. Le collage est introduit **manuellement** comme suit :

- Dans le ***CdC*** : expression du collage entre R-21 et R-8

ID	*Description*	*Event-B Model*
glue-R-21′-R-8	**[initiat] if [temperature] falls below 33C and [ap] falls below [lower_press_limit] then disconnect [dialyser], execute [alarm_temp], stop [BP] and execute [alarm_ap]**	R-21'_Ctx, R-21'_Mch

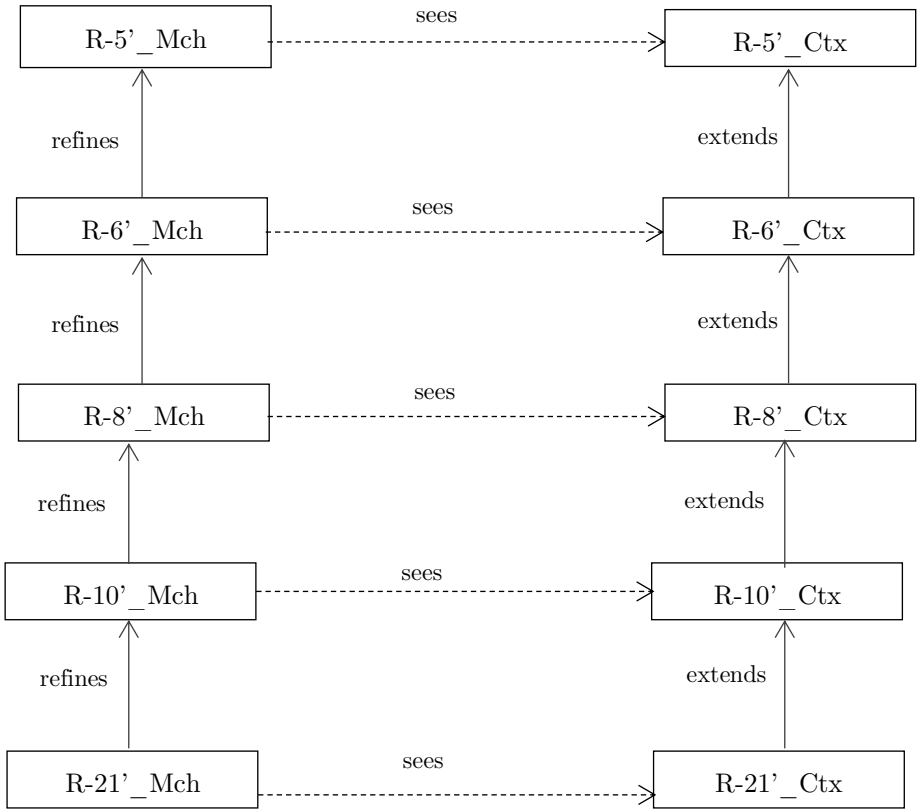
- Dans la ***Spec*** :
  - Invariant de collage

glue-R-21'-R-8: alarm_temp = ALM_deficit_temp ∧ alarm_ap = ALM_deficit_ap ⇒ temperature < 33 ∧ phase = initiat ∧ **ap < lower_press_limit** // invariant de collage

  - Événement de collage entre R-21 et R-8

Event **glue_treatment_R-21′_R-8'**
refines **treatment_R-8'**
when
grd1: phase = initiat
grd2: ap < lower_press_limit
grd3: temperature < 33
then
R-21′-act1: dialyser := disconnected ∧ alarm_temp := ALM_deficit_temp // action dans R-21
R-8'-act1: BP := stopped ∧ alarm_ap :=ALM_deficit_ap // action dans R-8

c) Schéma des Specs développées

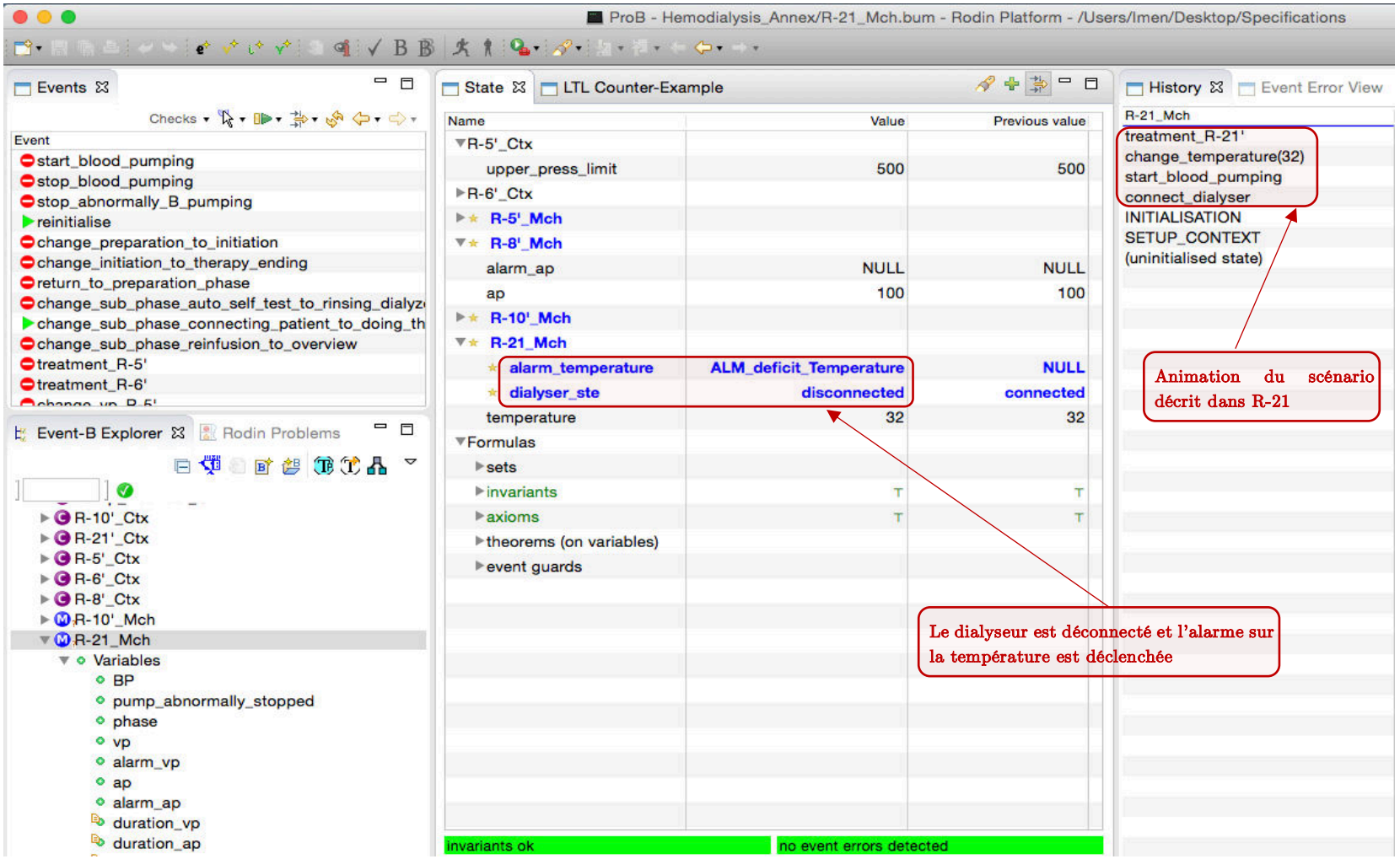


d) Vérification

Spec	***OPs automatiques***	***OPs interactives***	***Total***
R-5'_ Mch	21	0	21
R-6'_Mch	19	4	23
R-8'_Mch	36	2	38
R-10'_Mch	27	11	38
R-21'_Mch	21	7	28

e) Validation

Nous effectuons l'animation du scénario suivant : *connect_dialyser* → *start_blood_pumping* → *change_temperature (32)* → *treatment_R-21'* (voir figure de la page suivante).


ProB - Hemodialysis_Annex/R-21_Mch.bum - Rodin Platform - /Users/Imen/Desktop/Specifications
Events
Checks
Event
start_blood_pumping
stop_blood_pumping
stop_abnormally_B_pumping
reinitialise
change_preparation_to_initiation
change_initiation_to_therapy_ending
return_to_preparation_phase
change_sub_phase_auto_self_test_to_rinsing_dialyz
change_sub_phase_connecting_patient_to_doing_th
change_sub_phase_reinfusion_to_overview
treatment_R-5'
treatment_R-6'
Event-B Explorer
Rodin Problems
R-10'_Ctx
R-21'_Ctx
R-5'_Ctx
R-6'_Ctx
R-8'_Ctx
R-10'_Mch
R-21_Mch
Variables
BP
pump_abnormally_stopped
phase
vp
alarm_vp
ap
alarm_ap
duration_vp
duration_ap
State
LTL Counter-Example
Name
Value
Previous value
R-5'_Ctx
upper_press_limit
500
500
R-6'_Ctx
R-5'_Mch
R-8'_Mch
alarm_ap
NULL
NULL
ap
100
100
R-10'_Mch
R-21_Mch
alarm_temperature
ALM_deficit_Temperature
NULL
dialyser_ste
disconnected
connected
temperature
32
32
Formulas
sets
invariants
T
T
axioms
T
T
theorems (on variables)
event guards
invariants ok
no event errors detected
History
Event Error View
R-21_Mch
treatment_R-21'
change_temperature(32)
start_blood_pumping
connect_dialyser
INITIALISATION
SETUP_CONTEXT
(uninitialised state)
Animation du scénario décrit dans R-21
Le dialyseur est déconnecté et l'alarme sur la température est déclenchée

B.2.5.2 Prise en compte de R-21 relativement à R-8

Nous instancions Dev-if en prenant en compte le développement existant de R-8 en **B.2.3**. Nous partons d'un état du système où le développement de R-10 n'existe pas.

a) Comparaison entre ces deux besoins

	R-21	R-8
S_{p1}	If the machine is in the initiation phase	During initiation
S_{p2}	the temperature	the pressure at the AP transducer
S_{p3}	- the dialyser - an alarm signal	- the BP - an alarm signal
clause if	the temperature falls below the minimum temperature of 33C	the pressure at the AP transducer falls below the lower pressure limit
clause then	the software shall disconnect the dialyser from the DF and execute an alarm signal	the software shall stop the BP and execute an alarm signal

b) Instanciation de Dev-if sur R-21

CdC

ID	*Description*	*Event-B Model*
...	...	...
glue-R-8′-R-6	[initiat] if [ap] and [vp] fall below [lower_press_limit] then stop [BP] and execute [alarm_ap] and [alarm_vp]	R-8'_Ctx, R-8'_Mch
R-21'	**[initiat] if [temperature] falls below 33C then disconnect [dialyser] and execute [alarm_temp]**	**R-21'_Ctx, R-21'_Mch**
R-21′-init	**[temperature] is 37 [dialyser] is [disconnected]**	**R-21'_Ctx, R-21'_Mch**
R-21′-evolve	**[temperature] can [change_temperature]**	**R-21'_Ctx, R-21'_Mch**
glue-R-21'-R-8	**...**	

Glossaire

Formal term	*Informal description*
...	...
lower_press_limit	the lower pressure limit, the defined lower pressure limit
ap	the pressure at the AP transducer
alarm_ap	an alarm signal
change_ap	change the pressure at the AP transducer
temperature	**the temperature**
dialyser	**the dialyser**
alarm_temp	**an alarm signal**
disconnected	**disconnected**
change_temperature	**change the temperature**

Spec

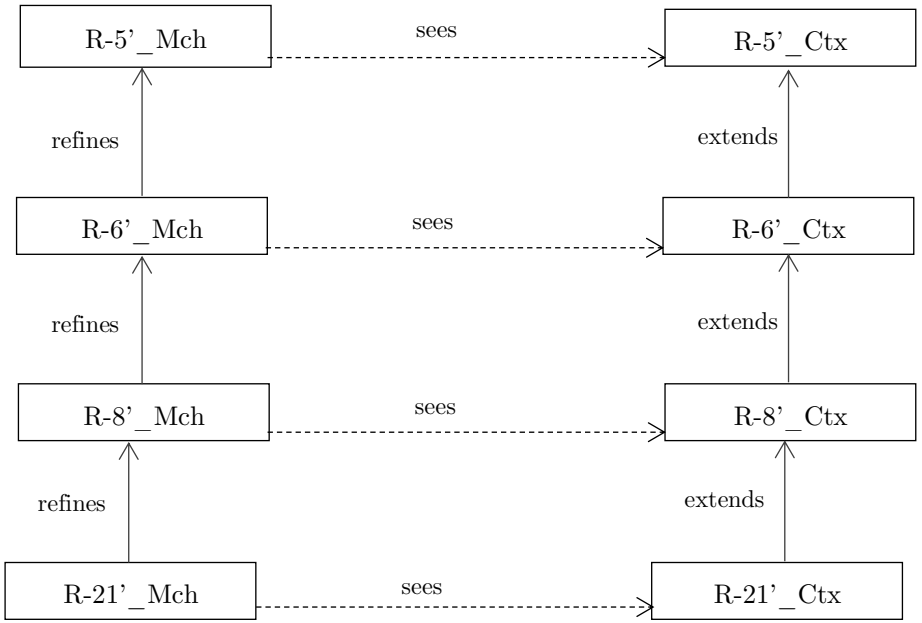
CONTEXT **R-21'_Ctx**
EXTENDS **R-8'_Ctx**
SETS
DIALYSER_STATE
CONSTANTS
connected, disconnected, ALM_deficit_temp
AXIOMS
axm31: partition (DIALYSER_STATE, {connected}, {disconnected})
axm32: partition (ALARMS, {ALM_deficit_temp})
END

MACHINE **R-21'_Mch**
REFINES **R-8'_Mch**
SEES **R-21'_Ctx**
VARIABLES
phase, vp, BP, alarm_vp, ap, alarm_ap, duration_vp
temperature, dialyser, alarm_temp
INVARIANTS
R-21'-temperature-Typ: temperature ∈ ℕ
R-21'-dialyser-Typ: dialyser ∈ DIALYSER_STATE
R-21'-alarm_temp-Typ: alarm_temp ∈ ALARMS
R-21'-normal: ¬(temperature < 33 ∧ phase = initiat) ⇒ ...
R-21'-anomaly: ... ⇒ temperature < 33 ∧ phase = initiat
glue-R-21′-R-8: ... ⇒ temperature < 33 ∧ phase = initiat ∧ ...
EVENTS
INITIALISATION
begin
R-21'-init-temperature: temperature := 37
R-21'-init-dialyser: dialyser := disconnected
End
Event **change_temperature**
Event **treatment_R-21'**
Event **glue_treatment_R-21' // apparaît avec l'instanciation de Dev-if**
Event change_ap // hérité du développement de R-8
Event treatment_R-8' // hérité du développement de R-8
Event treatment_R-6' // hérité du développement de R-6
...
END

NB.

- Contrairement au développement effectué en **B.2.5.1**, le **collage** entre R-21 et R-8 apparaît **immédiatement** et **automatiquement** dans ce développement.
- Le développement obtenu dans ce paragraphe est complété par le spécifieur. Nous obtenons les mêmes éléments de collage décrits dans la page 202.

c) Schéma des Specs développées



d) Vérification

Spec	*OPs automatiques*	*OPs interactives*	*Total*
R-5'_ Mch	21	0	21
R-6'_Mch	19	4	23
R-8'_Mch	36	2	38
R-21'_Mch	21	7	28

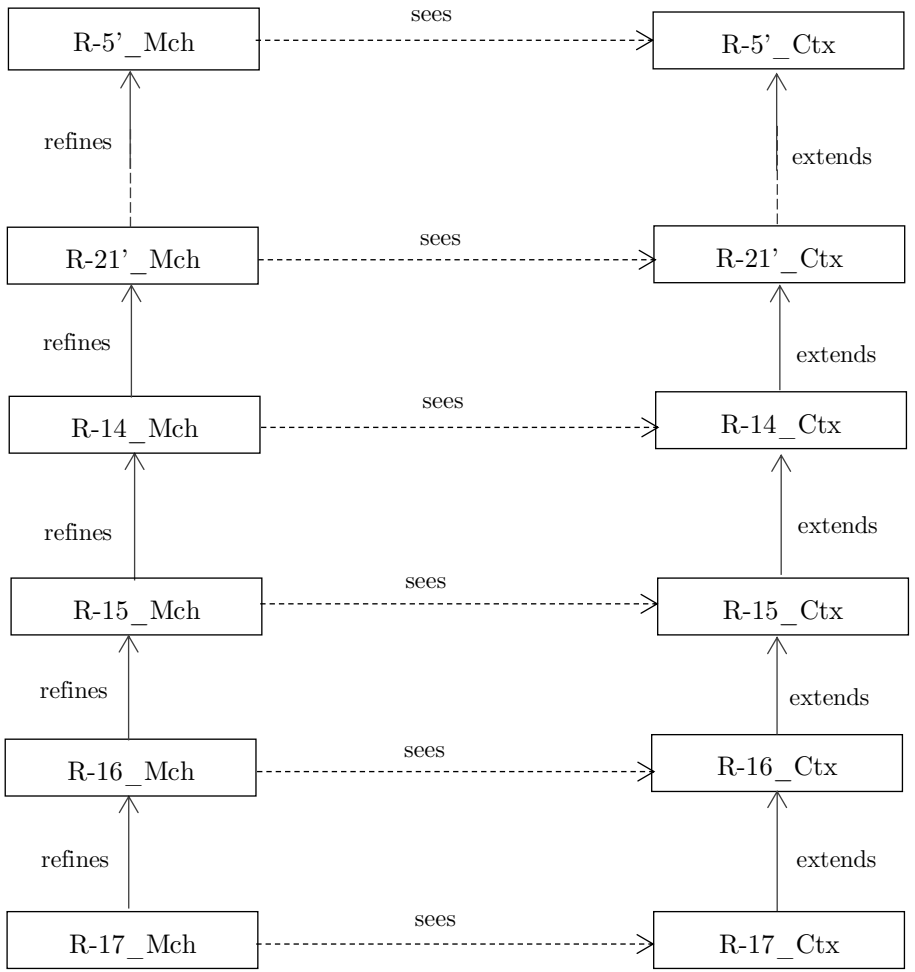
e) Validation

Elle est effectuée avec succès en animant le scénario suivant : *connect_ dialyser → start_ blood_ pumping → change_ temperature (32) → treatment_ R-21'.*

B.3 Evolution du développement

Nous avons développé les systèmes associés aux besoin *R-14*, *R-15*, *R-16* et *R-17*. Ces systèmes sont développés sans utilisation du patron Dev-if.

B.3.1 Schéma des Specs développées



B.3.2 Vérification

Spec	*OPs automatiques*	*OPs interactives*	
R-5'_ Mch	21	0	
R-6'_Mch	19	4	
R-8'_Mch	36	2	
R-10'_Mch	27	11	
R-21'_Mch	21	7	
R-14_Mch	25	0	
R-15_Mch	49	3	
R-16_Mch	30	4	
R-17_Mch	73	9	
Total	**301**	**40**	**341**

B.3.3 Validation

Au fur et à mesure du développement, nous avons validé toutes les Specs élaborées pour cette étude de cas. Nous avons utilisé l'outil ProB pour assurer cette activité.

B.4 Conclusion

Les besoins de la partie *« 4.2 Software requirements »* de cette étude de cas s'intéressent à la description des cas de faille des composantes de la machine d'hémodialyse. A chaque anomalie est associée une ou plusieurs réactions du système de contrôle de cette machine. Les besoins ont une forme conditionnelle identique. Nous avons pris en compte plusieurs besoins ayant cette forme pour développer semi automatiquement le système d'hémodialyse. Ce développement est guidé par l'utilisation du patron Dev-if. Les activités de vérification et de validation sont effectuées au fur et à mesure de l'élaboration du système grâce aux outils disponibles sous la plateforme Rodin. Les autres besoins, non encore pris en compte, seront développés de la même façon.