Etude de cas : modélisation de l'accès au laboratoire du Gloria

Alexandre Farnier, Gary Guyot, Yoni Levy, Romain Tissier

Janvier 2017

1 Introduction

L'objectif de ce projet est de modéliser les entrées/sorties de personnes au sein du célèbre laboratoire GLORIA. À l'aide de RODIN, il nous faut décrire l'ensemble des entités qui entrent en jeu dans le modèle, les contraintes qui s'y appliquent, ainsi que les différents raffinements nécessaires.

2 Modèles proposés

Suite à l'analyse du sujet, nous avons défini les trois grands points du modèle qui sont les données, les actions associées et les contraintes appliquées à ces données.

2.1 Données

Nous avons donc un ensemble de personnes, appelé "Personnes" par la suite, dans lequel on peut retrouver des visiteurs (Visiteurs), des employés (Employes) ainsi que des hôtesses (Hotesses). Étant donné que le sujet porte sur le Gloria, il a fallu créer un ensemble qui regroupe les personnes se trouvant dans le Gloria (InterieurGloria) ainsi qu'un autre ensemble qui est son complémentaire, les personnes se trouvant à l'extérieur de Gloria (ExterieurGloria).

Par la suite, nous avons vu que le Gloria possédait une "boite noire" qui fait office de dispositif de sécurité permettant de faire rentrer ou sortir des personnes avec une carte d'accès ou bien un code de sécurité. Nous avons donc créé les ensembles AUneCarte et AUnCode. Ces ensembles seront utiles par la suite dans les actions.

Enfin, le Gloria distingue deux types de personnes qui peuvent venir dans ses locaux : les Permanents et les Non permanents. Tout comme les ensembles des personnes se trouvant à l'intérieur et à l'extérieur de Gloria, ces deux ensembles sont complémentaires. Ces ensembles seront utilisés dans la suite afin de définir des contraintes sur les horaires d'accès au Gloria.

2.2 Actions

On peut distinguer cinq grandes actions dont les deux premières permettent d'entrer dans le bâtiment du Gloria, tandis que les trois autres permettent à une personne de sortir du bâtiment.

Tout d'abord, cette première action concerne toutes les personnes. En effet, si une personne souhaite rentrer au Gloria afin de voir une autre personne se trouvant déjà dans le bâtiment, et si une hôtesse est présente afin de vérifier son identité alors ce dernier pourra entrer dans le bâtiment.

La deuxième action concerne elle toutes les personnes possédants une carte et qui souhaite rentrer dans le bâtiment du Gloria. La carte fait office de pass et permet donc à toute personne en possédant une de rentrer dans le bâtiment si la boite noire lui autorise l'accès.

Pour sortir du batiment, on dispose de trois possibilités. La première est de se faire enregistrer par une hôtesse si elle est présente dans le bâtiment et sortir. La deuxième consiste à utiliser le code d'accès que l'on a obtenu sur la boite noite et sortir. Enfin la dernière permet à toute personne possédant une carte de sortir en utilisant sa carte sur la boite noire pour autoriser la sortie.

2.3 Contraintes

Pour les contraintes aussi on peut distinguer trois type de contraintes différentes. Le premier concerne uniquement les sorties du bâtiment du Gloria. En effet, si une personne à perdu sa carte ou son code d'accès et qu'il n'y a pas d'hôtesse pour enregistrer sa sortie, elle peut appeler un numéro de sécurité qui lui permettra de sortir du bâtiment. On a aussi le cas d'une urgence dans le bâtiment qui ouvre toutes les portes et qui permet à tout le monde de sortir du Gloria.

Le second lui utilise la distinction de permanents et de non permanent pour mettre en place des horaires d'accès au bâtiment. L'accès du Gloria est interdit à tout le monde avant les 7h ainsi qu'après 22h. Les non permanents eux ne peuvent accéder au batiment seulement entre 8h et 18h mais seulement les jours ouvrés (tous les jours sauf le week-end et les jours fériés).

Enfin, le dernier type de contrainte concerne le blocage des entrées et des sorties en même temps dans le cas d'une maintenance du bâtiment.

3 Choix de conception

3.1 Représentation du problème

Tous les ensembles définis dans la section précédente s'articulent autour des contraintes suivantes :

— $Visiteurs \subseteq Personnes$: un visiteur est une personne
— $Employes \subseteq Personnes$: un employé est une personne
— $Hotesses \subseteq Personnes$: une hôtesse est une personne
— $NonPermanents \subseteq Personnes$: les non permanents forment un sous ensemble de personnes
— $Permanents \subseteq Personnes$: les permanents sont des personnes
— $Hotesses \cap Employes = \emptyset$: les hôtesses ne sont pas des employés
— $Visiteurs \cap Employes = \emptyset$: les visiteurs ne sont pas des employés
— $Visiteurs \cap Hotesses = \emptyset$: les visiteurs ne sont pas des hotesses
— $Permanents \cap NonPermanents = \emptyset$: les ensembles permanents et non permanents sont disjoints
— $Permanents \cup NonPermanents = Personnes$: les ensembles permanents et non permanents composent la totalité de l'ensemble personne.
— $Visiteurs \neq \emptyset$: nous avons au moins un visiteur
— $Employes \neq \emptyset$: nous avons au moins un employé
— $Hotesses \neq \emptyset$: nous avons au moins une hotesse
— $Visiteurs \subseteq NonPermanents$: les visiteurs sont considérés comme des non permanents
— $Employes \subseteq Permanents$: les employés sont forcements des permanents
— $Hotesses \subseteq Permanents$: les hôtesses sont aussi des permanentes

On déclare aussi les variables suivantes :

— p : une personne
— a : une personne a
— b : une personne B
— h : une hôtesse
— $InGloriaP$: booléen à TRUE si la personne P est dans le Gloria
— $InGloriaB$: booléen à TRUE si la personne B est dans le Gloria
— $InGloriaA$: booléen à TRUE si la personne A est dans le Gloria
— $InGloriaH$: booléen à TRUE si la personne H est dans le Gloria
— $AUneCarteA$: booléen à TRUE si la personne A possède une carte
— $AUneCarteB$: booléen à TRUE si la personne B possède une carte
— $AUneCarteP$: booléen à TRUE si la personne P possède une carte
— $AUneCarteH$: booléen à TRUE si la personne H possède une carte
— $AUnCodeA$: booléen à TRUE si la personne A possède un code
— $AUnCodeB$: booléen à TRUE si la personne B possède un code

— $AUnCodeP$: booléen à TRUE si la personne P possède un code
— $AUnCodeH$: booléen à TRUE si la personne H possède un code
— $Maintenance$: représente si le système est en maintenance ou non
— $Temps$: représente la minute courante
— $JourOuvre$: représente si le jour courant est ouvré ou non

4 Tentative méthode ensembliste

4.1 Modèle de base

4.1.1 Contexte

Dans cette nouvelle modélisation, nous avons simplifié le contexte. Nous considérons désormais un ensemble de personne composé de Visiteurs, Employes, Hotesses, AUneCarte et AUncode respectant les axiomes suivant :
— $Visiteurs$ et $Employes$ forment les partitions de Personnes
— $Hotesses \subseteq Employes$: en effet, dans cette version, on considère que les Hotesses sont des employés contrairement au modèle précédent
— $AUneCarte \subseteq Personnes$
— $Employes \subseteq AUneCarte$: on choisi ici de dire que tous les employés possèdent une carte
— $AUnCode \subseteq Personnes$

Le contexte est donc défini comme suit dans Rodin :

CONTEXT
Control0
SETS
Personnes
CONSTANTS
Visiteurs
Employes
Hotesses
AUneCarte
AUnCode
AXIOMS
axm1 : $partition(Personnes, Visiteurs, Employes)$
axm2 : $Hotesses \subseteq Employes$
axm3 : $AUneCarte \subseteq Personnes$
axm4 : $Employes \subseteq AUneCarte$
axm5 : $AUnCode \subseteq Personnes$
END

4.1.2 Machine

Ce contexte est utilisé par une première machine simple mettant en oeuvre les variables suivantes :
— $InterieurGloria$: ensemble contenant les personnes étant à l'intérieur du Gloria
— $ExterieurGloria$: ensemble contenant les personnes étant à l'extérieur du Gloria
— $PersonneA$: une personne A quelconque
— $PersonneB$: une personne B quelconque
Ces variables suivent les invariants :
—
— $InterieurGloria \subseteq Personnes$: l'ensemble InterieurGloria est composé de Personnes

— $ExterieurGloria \subseteq Personnes$: l'ensemble ExterieurGloria est composé de Personnes
— $InterieurGloria \cap ExterieurGloria = \emptyset$: traduit qu'une personne ne peut pas être à la fois à l'intérieur et à l'extérieur du Gloria

Et sont initialisés de manière à ce que l'on considère le Gloria vide à l'état initial avec toute les personnes à l'extérieur.

Enfin, la machine décrit les observations décrites dans le sujet :

— Une personne A vient visiter une personne B et demande à l'hôtesse ses papiers et l'hôtesse enregistre l'identité de la personne et lui donne l'accès le cas échéant. (observation 1)
— Une personne A a une carte d'accès et l'utilise lors de son accès ; si l'accès lui est accordée, alors le dispositif d'accès est ouvert et la personne A peut entrer dans le bâtiment. (observation 2
— Une personne A veut sortir du bâtiment et l'hôtesse est présente ; l'hôtesse enregistre la sortie de cette personne qui sort du bâtiment. (observation 3)
— Une personne A veut sortir du bâtiment et l'hôtesse n'est plus présente ; A doit indiquer un code sur un dispositif contrôlant la sortie. (observation 4)
— Une personne a une accès par carte et doit présenter cette carte pour avoir l'autorisation de sortir. (observation 5)

La machine est donc définie comme suit dans Rodin :

MACHINE
ControlM0
SEES
Control0
VARIABLES
InterieurGloria
ExterieurGloria
PersonneA
PersonneB
INVARIANTS
inv1 : $InterieurGloria \subseteq Personnes$
inv2 : $ExterieurGloria \subseteq Personnes$
inv3 : $InterieurGloria \cap ExterieurGloria = \emptyset$
inv4 : $PersonneA \in Personnes$
inv5 : $PersonneB \in Personnes$

EVENTS
INITIALISATION $\triangleq$
STATUS
ordinary
BEGIN
act1 : $ExterieurGloria := Personnes$
act2 : $InterieurGloria := \emptyset$
act3 : $PersonneA :\in Personnes$
act4 : $PersonneB :\in Personnes$
END

Observation1 $\triangleq$
STATUS
ordinary
WHEN
grd1 : $PersonneA \in ExterieurGloria$
grd2 : $PersonneB \in InterieurGloria$
grd3 : $Hotesses \cap InterieurGloria \neq \emptyset$

THEN
act1 : $ExterieurGloria := ExterieurGloria \setminus PersonneA$
act2 : $InterieurGloria := InterieurGloria \cup PersonneA$
END

Observation2 $\triangleq$
STATUS
ordinary
WHEN
grd1 : $PersonneA \in ExterieurGloria$
grd2 : $PersonneA \in AUneCarte$
THEN
act1 : $InterieurGloria := InterieurGloria \setminus PersonneA$
act2 : $ExterieurGloria := ExterieurGloria \cup PersonneA$
END

Observation3 $\triangleq$
STATUS
ordinary
WHEN
grd1 : $PersonneA \in InterieurGloria$
grd2 : $Hotesses \cap InterieurGloria \neq \emptyset$
THEN
act1 : $InterieurGloria := InterieurGloria \setminus PersonneA$
act2 : $ExterieurGloria := ExterieurGloria \cup PersonneA$
END

Observation4 $\triangleq$
STATUS
ordinary
WHEN
grd1 : $PersonneA \in InterieurGloria$
grd2 : $PersonneA \in AUnCode$
grd3 : $Hotesses \cap InterieurGloria = \emptyset$
THEN
act1 : $InterieurGloria := InterieurGloria \setminus PersonneA$
act2 : $ExterieurGloria := ExterieurGloria \cup PersonneA$
END

Observation5 $\triangleq$
STATUS
ordinary
WHEN
grd1 : $PersonneA \in InterieurGloria$
grd2 : $PersonneA \in AUneCarte$
THEN
act1 : $InterieurGloria := InterieurGloria \setminus PersonneA$
act2 : $ExterieurGloria := ExterieurGloria \cup PersonneA$
END

END

4.2 Raffinement

4.2.1 Prise en compte des situations "urgence" et "maintenance"

Pour prendre en compte ces situations, nous avons étendu le contexte pour prendre en compte l'ensemble Situations composé des singletons Urgence, Maintenance et Normal.

CONTEXT
Control1
EXTENDS
Control0
SETS
Situations
CONSTANTS
Urgence
Maintenance
Normal
AXIOMS
axm1 : $partition(Situations, Urgence, Maintenance, Normal)$
END

Ensuite, nous avons raffiné notre machine afin de prendre en compte ces nouveaux éléments.

MACHINE
ControlM0
SEES
Control0
VARIABLES
InterieurGloria
ExterieurGloria
PersonneA
PersonneB
INVARIANTS
inv1 : $InterieurGloria \subseteq Personnes$
inv3 : $ExterieurGloria \subseteq Personnes$
inv4 : $InterieurGloria \cap ExterieurGloria = \emptyset$
inv5 : $PersonneA \in Personnes$
inv6 : $PersonneB \in Personnes$

EVENTS
INITIALISATION $\triangleq$
STATUS
ordinary
BEGIN
act1 : $ExterieurGloria := Personnes$
act2 : $InterieurGloria := \emptyset$
act3 : $PersonneA :\in Personnes$
act4 : $PersonneB :\in Personnes$
END

Observation1 $\triangleq$
STATUS
ordinary
WHEN
grd1 : $PersonneA \in ExterieurGloria$
grd2 : $PersonneB \in InterieurGloria$
grd3 : $Hotesses \cap InterieurGloria \neq \emptyset$
THEN
act1 : $ExterieurGloria := ExterieurGloria \setminus PersonneA$
act2 : $InterieurGloria := InterieurGloria \cup PersonneA$
END

Observation2 $\triangleq$
STATUS
ordinary
WHEN
grd1 : $PersonneA \in ExterieurGloria$
grd2 : $PersonneA \in AUneCarte$
THEN
act1 : $InterieurGloria := InterieurGloria \setminus PersonneA$
act2 : $ExterieurGloria := ExterieurGloria \cup PersonneA$
END

Observation3 $\triangleq$
STATUS
ordinary
WHEN
grd1 : $PersonneA \in InterieurGloria$
grd2 : $Hotesses \cap InterieurGloria \neq \emptyset$
THEN
act1 : $InterieurGloria := InterieurGloria \setminus PersonneA$
act2 : $ExterieurGloria := ExterieurGloria \cup PersonneA$
END

Observation4 $\triangleq$
STATUS
ordinary
WHEN
grd1 : $PersonneA \in InterieurGloria$
grd2 : $PersonneA \in AUnCode$
grd3 : $Hotesses \cap InterieurGloria = \emptyset$
THEN
act1 : $InterieurGloria := InterieurGloria \setminus PersonneA$
act2 : $ExterieurGloria := ExterieurGloria \cup PersonneA$
END

Observation5 $\triangleq$
STATUS
ordinary
WHEN
grd1 : $PersonneA \in InterieurGloria$

grd2 : $PersonneA \in AUneCarte$

THEN

act1 : $InterieurGloria := InterieurGloria \setminus PersonneA$

act2 : $ExterieurGloria : ExterieurGloria \cup PersonneA$

END

END

4.2.2 Prise en compte des horaires

Nous avons réalisé un second raffinement, pour prendre en compte les horaires, pour cela, nous avons de nouveau étendu le contexte. Nous utilisons donc un ensemble Horaire contenant les 3 horaires possibles.

CONTEXT

Control2

EXTENDS

Control1

SETS

Horaires

CONSTANTS

SeptHeureVingtDeuxHeure

SeptHeureHuitHeure

DixHuitHeureVingtDeuxHeure

AXIOMS

axm1 : $partition(Horaires, SeptHeureVingtDeuxHeure, SeptHeureHuitHeure, DixHuitHeureVingtDeuxHeure)$

END

5 Difficultés rencontrées

Durant le développement sous Rodin, nous nous sommes heurtés à plusieurs problèmes. Les premiers étaient la prise en main de l'outil car malgré les TP sur Rodin, nous avions du mal à utiliser cet outil. Il nous a fallu plusieurs heures de travail sur l'outil afin de savoir comment bien l'utiliser pour enregistrer nos modèles et faire les preuves. Rodin nous as aussi donné du fil à retordre car l'outil nous a affiché des erreurs qui n'existaient pas ou bien des demandes de démonstrations triviales (comme $\forall n \in N \implies n \geq 0$)

De plus, nous avons mis beaucoup de temps avant de commencer à développer sous Rodin. En effet, nous avons pris beaucoup de temps pour étudier le contexte du sujet, et essayer de trouver les différents éléments mettre en place.

6 Conclusion

Nous avons passé beaucoup de temps sur ce projet afin d'essayer de le mener à bien. Le travail sous Rodin a été la partie la plus longue du projet car nous avons recommencé à de nombreuses reprises la définition de notre contexte afin de le rendre le mieux possible. En ce qui concerne l'application, nous avons utilisé le plugin EventB2Java Generator afin de transformer les machines créées en programme Java. Nous n'avons cependant fait aucunes interprétations ou modélisation de ces fichiers Java.

Grâce à ce projet nous avons mieux compris le fonctionnement de Rodin mais aussi le concept de contexte et de machine abordé par Event B afin de modéliser des logiciels et applications sûrs de la vie de tous les jours.