ARTICLE 14 du réglement intérieur de l'N7 – PLAGIAT
Conformément au Code de la Propriété intellectuelle (articles L335-3, L112-1, L112-4), le plagiat
est un délit. Il se définit comme "toute reproduction, représentation ou diffusion, par quelque moyen que ce soit, d'une œuvre de l'esprit en violation des droits de l'auteur, tels qu'ils sont définis et réglementés par la loi". Il peut prendre différentes formes :
- oubli de référencer ses emprunts (extraits de texte, théories, sites internet, articles, thèses, images, tableaux de données, graphiques ...) ;
- paraphraser les idées d'un auteur sans citer ses sources d'inspiration ;
- traduction totale ou partielle d'une publication étrangère sans en indiquer la provenance ;
- appropriation de travaux pré-faits ou achetés sur Internet ;
De même, sont considérés comme du plagiat :
- l'utilisation de travaux d'autres étudiants avec ou sans leur consentement ;
- l'auto-plagiat (présenter le même travail déjà évalué, pour différentes années, matières ou établissement).
Il est précisé que l'INP est doté de logiciels "anti-plagiat" qui détectent toutes les parties et citations, utilisées dans une copie, issues d'internet et qui en retrouvent la source.
Tout cas de plagiat est passible d'une procédure relevant de la section disciplinaire décrite à l'article 15 du présent règlement de scolarité, les sanctions disciplinaires encourues peuvent aller jusqu'à l'interdiction de passer l'examen présenté pendant plusieurs années, mais également de sanctions administratives par la commission des fraudes, de sanctions civiles ou pénales selon le Code de l'Education et la loi du 23/12/1901 réprimant les fraudes.

Il est fortement probable que vous tombiez sur ce rapport après une recherche sur Github ou le partage du lien du répertoire, gardez à l'esprit que ce rapport ne constitue qu'une proposition de solution partiellement implémentée avec Event-B par 3 personnes très fatigués de l'N7.

Nous aurons déjà quittés l'N7 et ne préteront pas attention à toute sanction disciplinaire si vous recopiez bétement notre solution ou notre rapport. Connaissant plutôt bien tout professeur de l'équipe Acadie (méthodes formelles), il tient à votre **survie** de ne pas vous faire attraper. Nous avons tous déjà triché, repris le rapport des années précendentes, utilisé Chat-GPT, le but est de **ne pas se faire repérer.**
Nous avons écrit ce rapport en un temps où les détecteurs d'IA étaient peu fiables, rien ne dit que d'ici là, ils soient quasi-parfait.

Vous serez très sympathique de ne pas chercher à nous contacter, nous ne pourrons rien faire pour vous.

Cependant, si jamais les méthodes formelles vous intéressent, vous pouvez contacter Kilyan Le Gallic qui se fera un plaisir de vous aiguiller

Rapport - Développement Formel de Systèmes

Système de Gestion Ferroviaire

UE - Raffinement et Méthodes Formelles

Formation Sciences du Numérique - Parcours Logiciels - 3ème année

Table des matières

1. Introduction

Ce projet a pour objectif de réaliser une modélisation d'un système de gestion ferroviaire à l'aide d'Event-B. Un système de gestion ferroviaire s'assure de la circulation d'un ensemble de trains sur un réseau ferroviaire complexe en s'assurant de la sécurité des trains ainsi que de leur vivacité. Le modèle développé intègre des comportements et concepts complexes propres au domaine du ferroviaire comme la gestion des autorités de mouvements, les routes partielles et le comportement dynamique des trains.

La définition des contraintes nécessaires pour s'assurer de la correction du modèle développé a été découpée pour correspondre à chaque couche logique détaillée dans le cahier des charges, en différenciant hypothèse, contrainte fonctionnelle et contrainte de sécurité.

Le développement du système est effectué de manière itérative pour assurer la correction du modèle face aux contraintes exprimées, avec chaque couche logique entraînant plusieurs machines Event-B.

Le système devant être validé et prouvé, les activités de preuves et de *model-checking* seront détaillés dans la suite du rapport.

2. Définition des Exigences

Dans le but d'être en accord avec le cahier des charges, ainsi que pour s'assurer dès le départ que notre modélisation racine est correcte, nous avons défini les différentes exigences. Ceci nous a permis de modifier le moins possible ces exigences lors de la suite, bien que nous ayons eu malgré tout certaines modifications à apporter.

Nous avons découpé les différentes exigences suivant les trois couches logiques du sujet : Topologie et Signalisation (*TS*), Matériel Roulant (*MR*), Physique et Dynamique (*PHDY*). Les exigences sont elles mêmes découpées en trois sections : les besoins (*REQ*), les hypothèses (HYP), les sûretés (*SAF*). Chaque exigence possède un identifiant avec numéro à l'intérieur (le numéro est seulement présent pour différencier les exigences, il ne représente en aucun cas un ordre). De plus, les sûretés possèdent le chiffre du raffinage auquel il se rapporte. Par exemple, nous avons *REQ-MR-1* qui est l'identifiant du premier (*1*) besoin (*REQ*) de la couche Matériel Roulant (*MR*). De même, *SAF-TS-0-1* est l'identifiant de la première (*1*) sûreté (*SAF*) du premier raffinage (*0*) de la couche Topologie et Signalisation (*TS*).

Nous avons choisi de faire la définition des exigences sous forme de tableaux. Ceux-ci comportent quatre colonnes : *Identifiant*, *Exigence*, *Modélisée*, *Commentaire (facultatif)*. La première colonne permet de savoir à quelle exigence nous nous reportons, notamment lorsque nous la mettons en place dans le code. La seconde colonne est l'exigence en elle-même.

2.1. Topologie et Signalisation

2.1.1. Exigences générales

2.1.1.1. Besoins

Voici nos exigences pour les besoins de Topologie et Signalisation. Les exigences qui demandent plus de détails sont décrites dans la partie Particularité.

Identifiant	Exigence
REQ-TS-1	Les trains se déplacent sur le réseau
REQ-TS-2	Un train en attente de route doit ◇ recevoir une route
REQ-TS-3	Un arrêt est composé d'une entrée et d'une sortie (deux tronçons différents)
REQ-TS-4	Un train en attente de route se met en mouvement dès réception d'une route si possible
REQ-TS-5	Une route est composée d'une suite de tronçons connexes
REQ-TS-6	Un train finissant sa route n'a plus de route

2.1.1.2. Hypothèses

Voici nos exigences pour les hypothèses de Topologie et Signalisation. Les exigences qui demandent plus de détails sont décrites dans la partie Particularité.

Identifiant	Exigence
HYP-TS-1	Le réseau se compose de tronçons connexes
HYP-TS-2	Un tronçon peut accueillir la totalité d'un train
HYP-TS-3	Une route est donnée par un algorithme parfait
HYP-TS-4	Une route ne peut pas faire demi-tour à un train
HYP-TS-5	Les tronçons sont disposés tel qu'à être parcourables par un train
HYP-TS-6	Si un tronçon est voisin d'un autre, alors l'inverse est vrai
HYP-TS-7	Un tronçon ne peut pas être voisin de lui-même
HYP-TS-8	Un arrêt possède une entrée et une sortie (deux tronçons)
HYP-TS-9	Un arrêt possède au moins une voie
HYP-TS-10	Les voies d'un arrêt sont voisins avec les deux tronçons d'entrée et sortie de ce même arrêt
HYP-TS-11	Deux arrêts peuvent uniquement se partager des voies d'entrée et de sortie, aucune voie d'un arrêt a1 ne doit servir de voie ou d'entrée sortie à un autre arrêt a2
HYP-TS-12	Une route entre deux arrêts débute par la sortie du premier arrêt et se termine sur l'entrée du second arrêt

2.1.1.3. Sûretés

Voici nos exigences pour les sûretés de Topologie et Signalisation. Les exigences qui demandent plus de détails sont décrites dans la partie Particularité.

Identifiant	Exigence
SAF-TS-0-1	Il ne doit pas y avoir deux trains dans un même tronçon
SAF-TS-0-2	Si un train existe alors il est sur un tronçon
SAF-TS-0-3	Un train sur un tronçon T ne peut avancer que sur un tronçon voisin de T
SAF-TS-0-4	Un train est au maximum dans 2 tronçons
SAF-TS-0-5	Si deux tronçons accueillent le même train alors ils sont connexes
SAF-TS-1-1	Un train peut avoir au maximum 1 arrêt en destination
SAF-TS-1-2	Un train sans destination est sur une voie d'un arrêt
SAF-TS-1-3	Un train dans le tronçon d'entrée d'un arrêt étant sa destination doit arriver dans une voie de l'arrêt
SAF-TS-1-4	Un train dans une voie d'un arrêt étant sa dernière destination ne peut pas en sortir tant qu'il n'a pas de nouvelle destination
SAF-TS-1-5	Si un train sort d'un arrêt, alors il a une nouvelle destination

SAF-TS-2-1	Un train sans route n'avance pas tant qu'il n'a pas de nouvelle route. Sauf pour occuper l'arrêt qui est sa destination ou sortir d'un arrêt qui était sa précédente destination
SAF-TS-2-2	Une route envoyée à un train débute par le tronçon actuel du train

2.1.2. Particularités

Lors de la définition des exigences, nous avions écrit les deux suivantes en tant que sûreté :

- *Si un tronçon est voisin d'un autre, alors l'inverse est vrai.*
- *Un tronçon ne peut pas être voisin de lui-même.*

Pendant l'écriture de la modélisation, nous nous sommes orientés sur la création des tronçons directement dans le contexte avec des constantes. Ainsi, les tronçons vont forcément avoir ces exigences étant donné que nous les créons nous mêmes. Par conséquent, ces exigences ont finalement été déplacées dans les hypothèses.

Pour les sûretés du raffinement 1, nous les avons pour la plupart modifiées ou ajoutées après la modélisation. En effet, nous pouvons remarquer dans la partie sur la modélisation que nous avons introduit la notion de voie. Une voie étant un tronçon lié à un arrêt, elle permet à un train de rester sur l'arrêt sans que les autres trains ne soient bloqués. D'ailleurs, cette nouvelle notion a agrandi la sûreté SAF-TS-2-1. En effet, les trains pouvant être sur les voies d'un arrêt, ceux-ci doivent pouvoir en sortir sans route, afin d'être en accord avec la sûreté SAF-TS-1-4.

2.2. Matériel Roulant

2.2.1. Exigences générales

Les exigences ne concernant pas les wagons sont en réalité des hypothèses ou sont déjà couvertes dans la modélisation des raffinages précédents. Ainsi, l'ajout des wagons et leurs propriétés consiste en une extension du comportement d'un train pour prendre en compte la position des wagons et la manière dont ils doivent se déplacer par rapport au train.

2.2.1.1. Besoins

Voici nos exigences pour les besoins de Topologie et Signalisation. Les exigences qui demandent plus de détails sont décrites dans la partie Particularité.

Identifiant	Exigence
REQ-MR-1	Il doit y avoir une unique motrice par train
REQ-MR-2	Un train ayant un wagon se trouvant sur un tronçon doit occuper le dit tronçon
REQ-MR-3	Les wagons changent de tronçon dans leur ordre d'arrimage sur le train

2.2.1.2. Hypothèses

Voici nos exigences pour les hypothèses de Topologie et Signalisation. Les exigences qui demandent plus de détails sont décrites dans la partie Particularité.

Identifiant	Exigence
HYP-MR-1	Le train est composé de wagons connexes
HYP-MR-2	Le train est insécable
HYP-MR-3	La taille d'un train est le nombre de ses wagons
HYP-MR-4	Un train ne peut pas déplacer plus de N wagons en un déplacement (N étant sa taille)

2.2.1.3. Sûretés

Voici nos exigences pour les sûretés de Topologie et Signalisation. Les exigences qui demandent plus de détails sont décrites dans la partie Particularité.

Identifiant	Exigence
SAF-MR-0-1	Les wagons d'un train se déplacent de manière uniforme et simultanée
SAF-MR-0-2	Un train a une taille positive et non nulle
SAF-MR-0-3	Si un train est sur deux tronçons, la somme des wagons sur les tronçons est égal à la taille du train
SAF-MR-0-4	Si un train est sur un seul tronçon, le nombre de wagons sur le tronçon est égal à la taille du train

2.2.2. Particularités

Le cahier des charges indique que dans cette couche un train ne peut être que sur un ou deux tronçons. Cette exigence a été couverte dans la modélisation de la machine racine, c'est-à-dire la couche “Topologie et Signalisation”, en tant que *SAF-TS-0-4*.

2.3. Physique et Dynamique

2.3.1. Exigences générales

2.3.1.1. Besoins

Voici nos exigences pour les besoins de Physique et Dynamique. Les exigences qui demandent plus de détails sont décrites dans la partie Particularité.

Identifiant	Exigence
REQ-PHDY-1	La longueur d'un train est la différence entre la position de sa tête et celle de sa queue
REQ-PHDY-2	L'autorité de mouvement d'un train est mise à jours à mesure que le train avance
REQ-PHDY-3	Un train est caractérisé par sa vitesse et son accélération
REQ-PHDY-4	Un freinage est une accélération négative
REQ-PHDY-5	Un train a une capacité de freinage limité
REQ-PHDY-6	Un train en mode Libre avance librement sur le réseau avec a quelconque
REQ-PHDY-7	Un freinage est une accélération négative
REQ-PHDY-8	Un train en Freinage peut repasser en Libre si SD < EoA
REQ-PHDY-9	Une autorité de mouvement se termine par une EoA

2.3.1.2. Hypothèses

Voici nos exigences pour les hypothèses de Physique et Dynamique. Les exigences qui demandent plus de détails sont décrites dans la partie Particularité.

Identifiant	Exigence
HYP-PHDY-1	La longueur d'un train est fixe
HYP-PHDY-2	Le système est régi par un temps discret de pas Δt fixe
HYP-PHDY-3	La distance de freinage d'un train est supposée donnée par un algorithme parfait
HYP-PHDY-4	La vitesse d'un train est la dérivée dp/dt
HYP-PHDY-5	L'accélération d'un train est la dérivée dv/dt

2.3.1.3. Sûretés

Voici nos exigences pour les sûretés de Physique et Dynamique. Les exigences qui demandent plus de détails sont décrites dans la partie Particularité.

Identifiant	Exigence
SAF-PHDY-0-1	Un train ne peut pas dépasser la fin de son autorité de mouvement
SAF-PHDY-0-2	Un train ne peut pas reculer
SAF-PHDY-0-3	Un train a une vitesse bornée par la constante V_max
SAF-PHDY-0-4	Un train a une accélération bornée par la constante A_max
SAF-PHDY-0-5	Un train en Freinage a une accélération négative et une vitesse décroissante
SAF-PHDY-0-6	Un train en Attente possède une vitesse/accélération nulle
SAF-PHDY-0-7	Un train en Libre passe en Freinage dès que SD = EoA
SAF-PHDY-0-8	Un train en Freinage passe en Attente dès que v=0
SAF-PHDY-0-9	Un train en Attente peut repasser en Libre si sa MA lui permet
SAF-PHDY-0-10	Si plusieurs trains doivent se retrouver sur le même tronçon en même temps, une ou plusieurs EoA doivent être placés de manière à éviter la collision

2.3.2. Particularités

Cette partie n’a pu être faite dans le code. Ainsi, nous n’avons pas changé d'exigence ici.

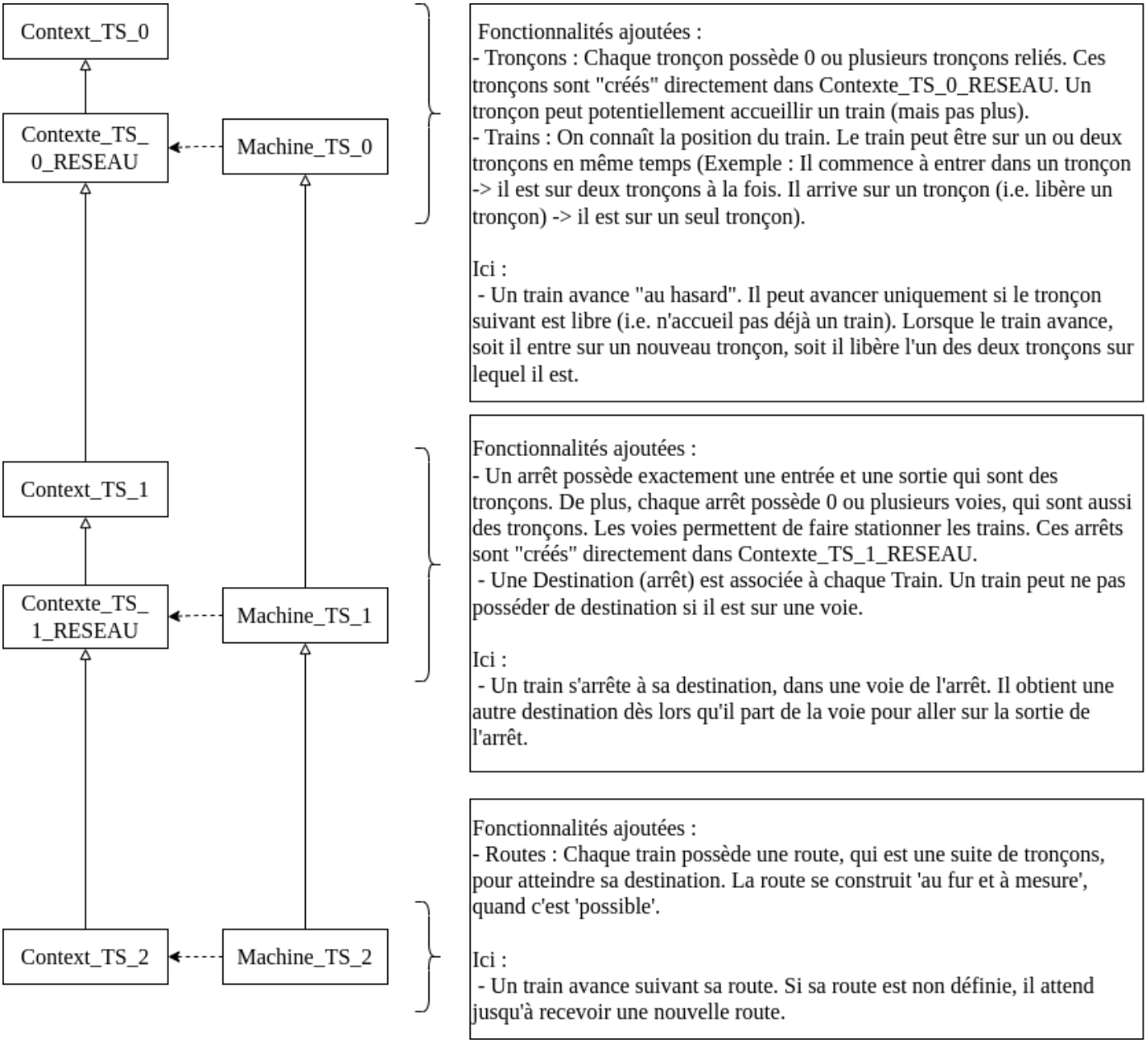
3. Modélisation

Suite à la réalisation des exigences, nous avons pu modéliser notre système de gestion ferroviaire. Cette partie est celle qui précède le code.

Comme pour la partie exigence, nous avons cherché à modéliser le mieux possible notre système de gestion ferroviaire dès le départ, afin de nous faciliter la suite. Dans cette même optique, nous étions totalement ouverts à tout changement pouvant s'avérer utile.

La modélisation s'est sectionnée en trois couches, comme indiqué dans le sujet : Topologie et Signalisation (*TS*), Matériel Roulant (*MR*), Physique et Dynamique (*PHDY*). Chacune de ces couches possèdent elles-mêmes plusieurs raffinages lorsque cela est nécessaire. Dans le but de garder la trace de nos réflexions, nous avons créé des diagrammes qui illustrent la modélisation. Ceux-ci se composent des fichiers et de la hiérarchie des fichiers sur la partie gauche, ainsi que des spécifications sur la partie droite. Le nom des fichiers indique leur type (*Context* ou *Machine*), la couche ciblée (*TS*, *MR*, *PHDY*) et le niveau de raffinage (en partant de 0).

3.1. Topologie et Signalisation



Topologie des raffinements concernant la couche “Topologie et Signalisation”

L’objectif de la couche logique “Topologie et Signalisation” est d’expliciter les concepts de routes, d’arrêts et la place générale des trains sur un réseau de tronçons connexes.

3.1.1. Création des tronçons

Lors de la première modélisation, nous avions défini que les tronçons seraient créés dynamiquement dans le code de la “machine”. Ceci fait que nous n’aurions donc pas connaissance des tronçons existant avant le lancement du code.

Nous nous sommes ainsi orientés dans la réalisation du code pour le premier raffinage. Cependant, avec l'appui de notre professeur, nous nous sommes rendus compte de la complexité que nous aurions ajoutée involontairement à notre machine. En effet, en plus d’avoir un code plus complexe à écrire, celui-ci serait également plus difficile à prouver par la suite.

Par conséquent, et sous les conseils avisés de notre professeur, nous avons choisi de faire la création des tronçons directement dans un contexte. Ainsi, nous connaissons clairement les tronçons et leurs liaisons. De plus, ceux-ci suivent forcément nos hypothèses, étant donné que c’est nous qui les créons.

Context_TS_2 n’apporte rien en soit, mais permet de garder une homogénéité dans le code. Nous avons estimé que rajouter un Context_TS_2_RESEAU pour garder cette homogénéité serait de trop.

3.1.2. Modélisation du voisinage des tronçons

Nous avons pu être amenés à avoir une réflexion sur la modélisation du voisinage des tronçons. Au départ, nous pensions représenter les voisins de droite, et les voisins de gauche séparément. Finalement, nous avons choisi de ne pas différencier les voisins.

Ce choix vient de plusieurs faits :
Premièrement, nous considérons que les tronçons, et donc leurs liaisons, sont fournis par un tiers, et que c’est celui-ci qui décide de leur positionnement. Ainsi, on met en hypothèse que les tronçons possèdent un positionnement cohérent et utilisable pour un train.
Ensuite, la route est considérée parfaite. Par conséquent, cette dernière connaîtra les tronçons à parcourir, et donc le tronçon suivant est forcément bien positionné (droite ou gauche).
Enfin, nous n’avons pas de cas où il nous serait utile de différencier les tronçons de droite, et ceux de gauche.

Ainsi, un tronçon pourra posséder, ou non, une liste de un ou plusieurs tronçons qui sont ses voisins.

3.1.3. Entrer / Arriver

Dans notre modélisation, un train avance de moitié de tronçon en moitié de tronçon. Ainsi, si un train est présent sur un seul tronçon, il va pouvoir *entrer* dans un nouveau tronçon. Et lorsque le train est sur deux tronçons, il peut *arriver* sur l’un des deux tronçons, soit quitter l’autre tronçon. Nous avons choisi de garder les mots clés ENTRER et ARRIVER, afin de se référer au même tronçon sur lequel on entre et on arrive.

3.1.4. Arrêts

Dans la réalité, un train est censé pouvoir stationner dans un arrêt, sans forcément vouloir repartir. Dans cette vision-là, nous avons fait le choix de posséder des tronçons au sein des arrêts. Ainsi, un train étant dans un arrêt et n'ayant pas de destination, ne bloquera pas un autre train. C'est alors que nous avons créé des voies, qui sont liées à des arrêts. Chaque arrêt possède 0 ou plusieurs voies. De plus, dans la continuité de la création du réseau, nous avons décidé de mettre la disposition des voies dans le contexte directement.

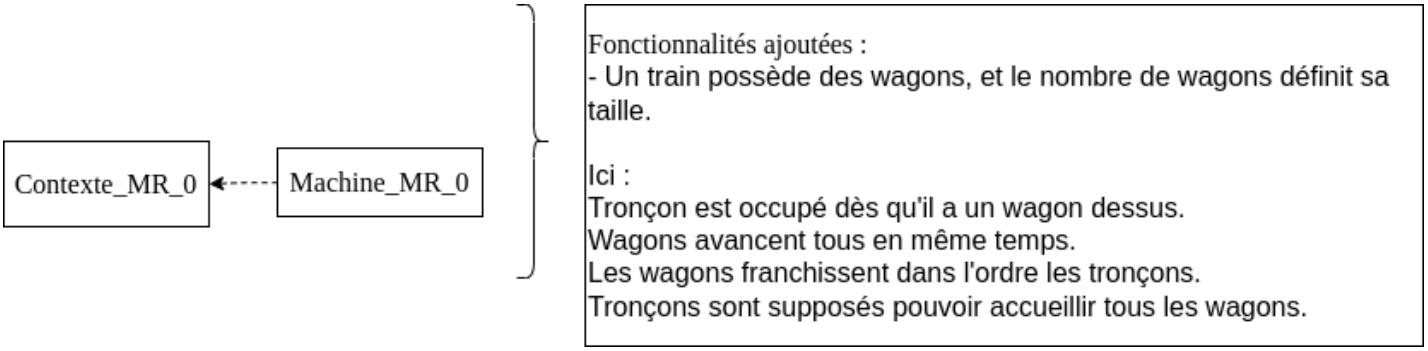
Comme pour la modélisation du voisinage des tronçons, nous considérons qu'un arrêt possède exactement une entrée et une sortie, mais nous ne définissons pas laquelle est laquelle. Les deux peuvent faire office d'entrée et de sortie.

Le cahier des charges indique qu'un train obtient une destination lorsqu'il est sur la sortie d'un arrêt. Par conséquent, l'événement *PARTIR* a été créé comme un raffinement de *ENTRER*. Cet événement va alors, en plus d'entrer sur le tronçon de sortie, ajouter une destination au train.

3.1.5. Routes

Dans notre modélisation, nous avons défini trois événements spécialement pour la route. Ceux-ci sont sa création (*CREER_ROUTE*), sa mise à jour (*RALLONGER_ROUTE*) et sa suppression (*SUPPRIMER_ROUTE*). La création de la route doit se faire avec l'un des tronçons actuels du train. Sa mise à jour, qui est l'ajout d'un tronçon, doit se faire avec un tronçon voisin à l'un des tronçons de la route, et qui n'est pas déjà dessus. Les tronçons de la route sont également supprimés lorsque le train sort du tronçon.

3.2. Matériel roulant

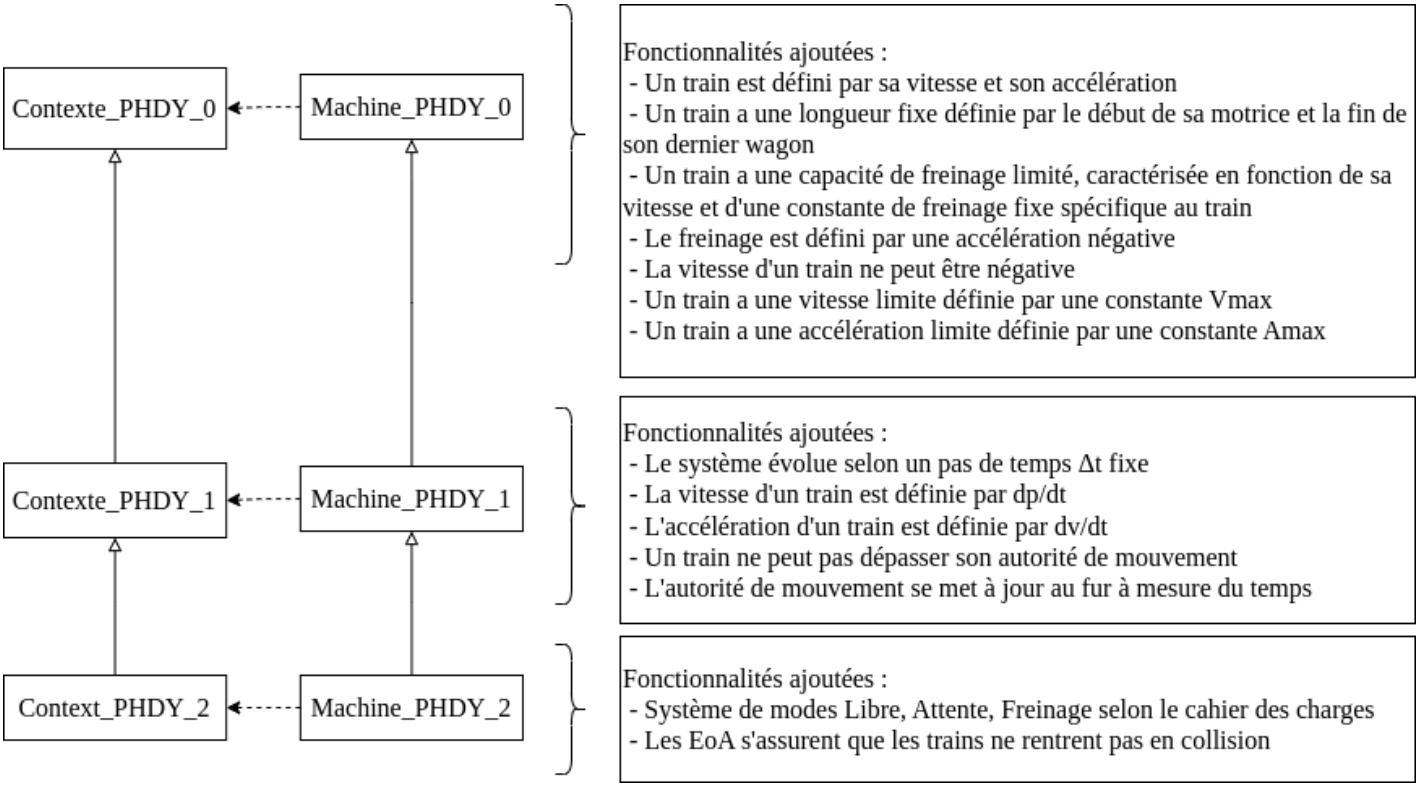


Topologie du raffinement concernant la couche “Matériel roulant”

L’objectif de la couche logique “Matériel roulant” est de gérer le train de manière indépendante en tant qu’assemblage de wagons non élastique situé sur un ou deux tronçons.

La modélisation que nous avons choisie pour cette partie est de prendre en compte les wagons comme une valeur de type entier numérique, en effet, les wagons sont présents dans deux éléments : la taille du train et le nombre de wagons contenus par un tronçon. Ce choix a été fait suite à une version alternative du raffinement Matériel Roulant qui présentait les wagons comme provenant d'un ensemble WAGONS. Nous avions plusieurs variables qui nous permettaient de les gérer tels que : trainPossedeWagon, wagonDerriereWagon, tronconPossedeWagon, numeroWagon. Or, nous avons été bloqué à l’étape du raffinement des fonctions de déplacement du train, nous étions dans l’incapacité de déplacer plus d’un wagon à la fois, enfin, le seul moyen pour y parvenir était d’utiliser l’affectation “:|” ce qui ne nous permettait d’avoir qu’un seul et unique train (car les informations sur les wagons des autres trains disparaissaient). Cela nous a donc amené à poursuivre cette étape de raffinement sans représenter les wagons en termes d’éléments d’un ensemble mais en une valeur numérique.

3.3. Physique et Dynamique



Topologie des raffinements concernant la couche “Physique et Dynamique”

L’objectif de la couche logique “Physique et Dynamique” est de caractériser un train en tant qu’objet mécaniquement dynamique décrit par une vitesse et une accélération et le système ferroviaire comme système évoluant en temps discret. Cette couche est également responsable de l’introduction du système d’autorités de mouvements et des modes des trains sur le réseau ferroviaire.

Cette partie n’a pas été réalisée dans le code, or, nous y avons quand même réfléchi. Effectivement, à la lecture des exigences de cette partie, nous trouvions assez évident de créer une fonction etatTrain qui affectait à un train un état qui appartient à un SET défini tel que ETATS est une partition de “Avance”, “Freine”, “Attend”. De plus, les événements permettant le mouvement des trains verraient des gardes s’ajouter pour être en cohérence avec l’état du train, par exemple : un train ne peut pas entrer dans un nouveau tronçon alors qu’il attend.

4. Preuves et animations avec Pro-B

4.1. Preuves

Par manque de connaissances, nous avons décidé de prioriser le rendu du rapport et du modèle sous Rodin. Ainsi, nous n'avons pas déchargé la totalité des obligations de preuves générées par Rodin pour notre modèle. En effet, elles se trouvent être relativement complexes et nous ne savons tout simplement pas les résoudre.

Les machines TS_1 et TS_2 voient leurs obligations de preuves entièrement déchargées à l'aide des solveurs SMT intégrés à Rodin.

4.2. Vérifications des modèles

La vérification des modèles a été faite au fur et à mesure en testant différents scénarios à l'aide de Pro-B. Vous trouverez ici une liste non exhaustive des scénarios que nous avons pu tester pour les différents raffinements des différentes modélisations.

4.2.1. Topologie et Signalisation

4.2.1.1. Raffinement 0

Scénarios :

- Vérification à toutes les étapes/scénarios :
    - Tous les trains sont sur au moins un tronçon, et au maximum sur deux.
    - Lorsqu'un train a la possibilité d'avancer, le tronçon sur lequel il peut aller est connexe à l'un des tronçons sur lequel il est.
    - Lorsqu'un train est sur deux tronçons, les deux sont connexes l'un à l'autre.
- Deux trains à la suite. L'un ne peut pas aller sur le·les tronçon·s de l'autre.

4.2.1.2. Raffinement 1

Scénarios :

- Vérification à toutes les étapes/scénarios :
    - Un train possède zéro ou un tronçon.
    - Si un train ne possède pas de destination, il est sur une voie d'un arrêt.
- Faire arriver le train à sa destination. Lorsque le train est dans un tronçon de l'arrêt de sa destination, son seul mouvement possible est d'arriver dans une voie de l'arrêt. Il n'est alors que sur un seul tronçon, qui est une voie de l'arrêt. Sa destination disparaît.
- Une fois que le train est dans sa dernière destination, il peut en sortir et il obtient bien une nouvelle destination.

4.2.1.3. Raffinement 2

Scénarios :

- Vérification à toutes les étapes/scénarios :
    - Un train sans route n'avance pas tant qu'il n'a pas de nouvelle route. Sauf pour occuper l'arrêt qui est sa destination ou sortir d'un arrêt qui était sa précédente destination.
- Créer une route pour un train, et vérifier que celle-ci débute par l'un des tronçons sur lequel il est.
- Sortir un train de la voie d'un arrêt. Vérifier que la destination n'est pas l'arrêt occupé précédemment.
- Sortir un train de la voie d'un arrêt. Créer une route pour un train, et vérifier que celle-ci débute soit par la sortie de l'arrêt soit par le tronçon de la voie (si le train fait demi-tour, ce n'est pas interdit).

4.2.2. Matériel roulant

4.2.2.1. Raffinement 0

Lors de ce raffinement, nous nous sommes rendus compte d'un problème. En effet, un train composé uniquement d'une motrice ne correspond pas aux gardes des événements définis. Malheureusement, le problème n'a pas pu être résolu et nous tiendrons uniquement compte de scénarios ou les trains ont une taille supérieure à 1.

Scénarios :

- Vérification à toutes les étapes/scénarios :
    - Les wagons d'un train passent la jonction entre deux tronçons dans leur ordre d'arrimage
    - Les wagons d'un train se trouvent sur un tronçon occupé par le train
    - La somme des wagons arrimés à un train sur un/deux tronçon(s) est égale à la taille du train
    - Un déplacement ne peut déplacer plus de N wagons pour un train de taille N
- Faire circuler un train jusqu'à son entrée sur un arrêt et vérifier que tous les wagons du train se trouve sur la voie de stockage de l'arrêt.
- Faire circuler un train pour faire entrer entre deux tronçons et vérifier que les wagons se trouvent de manière ordonnée sur les deux tronçons.
- Sortir un train d'un arrêt et vérifier qu'aucun wagon ne reste sur la voie de stockage.

4.2.3. Physique et Dynamique

4.2.3.1. Raffinement 0

Scénarios :

- Vérification à toutes les étapes/scénarios :
  - Un train ne peut avoir une vitesse négative.
  - Un train ne peut dépasser une certaine vitesse Vmax.
  - Un train ne peut dépasser une certaine accélération Amax.
  - La longueur d'un train reste fixe
- Faire accélérer un train jusqu'à sa vitesse Vmax avec une accélération de Amax.
- Faire freiner un train avec un accélération négative maximale jusqu'à arrêt et vérifier que le train ne recule pas.

4.2.3.2. Raffinement 1

Scénarios :

- Vérification à toutes les étapes/scénarios
  - L'autorité de mouvement d'un train évolue constamment.
  - Le système évolue selon un pas de temps Δt fixe.
  - L'accélaration et la vitesse d'un train sont respectivement caractérisées par la dérivée seconce et la dérivée première de la distance sur le temps.
  - Un train ne dépasse jamais son autorité de mouvement
- Faire circuler un unique train jusqu'à la fin de son autorité de mouvement et vérifier que sa vitesse s'adapte à la fin de son autorité de mouvement.
- Faire circuler plusieurs trains de manière à ce que l'autorité de mouvement d'un des trains en circulation soit restreinte subitement et vérifier que le train associé adapte sa vitesse.
- Faire circuler plusieurs trains sur le même tronçon et vérifier que les autorités de mouvements s'ajustent pour éviter une collision.

4.2.3.3. Raffinement 2

Scénarios :

- Vérification à toutes les étapes/scénarios :
  - Le mode d'un train appartient à la partition de l'ensemble {Attente, Freinage, Libre}
  - Un train respecte les spécifications des modes :
    - Freinage : Vitesse décroissante et accélération négative, passe en Libre si Autorité de Mouvement > Distance d'arrêt, passe en Attente dès que vitesse nulle.
    - Libre : Vitesse positive et accélération nulle ou positive, passe en Freinage si Autorité de Mouvement ≤ Distance d'arrêt.
    - Attente : Vitesse et accélération nulle, passe en Libre si Autorité de Mouvement mise à jour.
- Faire circuler un train jusqu'à son arrêt et vérifier qu'il est en mode Attente.

- Faire circuler un train jusqu'à la fin de son autorité de mouvement et vérifier qu'il est en mode Freinage avant la fin de son autorité de mouvement et en mode Attente sur la fin de son autorité de mouvement.
- Faire circuler plusieurs trains sur la même portion de tronçon jusqu'à un arrêt et vérifier que les trains passent successivent du mode Libre au mode Freinage au mode Attente.

5. Bilans personnels

5.1. REDACTED n°1

De manière générale, la matière Développement Formel de Systèmes m'a fortement intéressé, d'autant plus la partie pratique. En effet, lors de ce projet, j'ai été chargé de la réalisation des modèles en Event-B. J'ai apprécié effectuer ce travail d'autant plus avec des outils comme ProB, en revanche, je ne ferai aucun commentaire sur le logiciel Rodin utilisé avec un pavé tactile… Malheureusement, je trouve qu'en tant qu'alternant la partie relative aux preuves m'est hors d'atteinte. Mise à part les déchargement automatique des obligations de preuves, je suis incapable de déterminer si la preuve est réalisable ou si le modèle n'est pas prouvable. J'aurais bien aimé ajouter ce mode de développement à mon arc de compétence mais je me retrouve bloqué par cet aspect là. En conclusion, j'ai tout de même bien aimé le projet, assez intéressant et complexe pour réfléchir et penser à de nombreux aspects différents ou contraintes à prendre en compte.

5.2. REDACTED n°2

La réalisation d'un projet en méthodes formelles m'a permis de mettre en œuvre les compétences acquises dans ma formation dans le cadre d'un travail de plus grande échelle qu'un TP.

Dans le cadre du projet, je suis responsable de la décharge des obligations de preuves et la vérification des modèles face à des scénarios élaborés de manière commune.

J'ai également participé à l'élaboration et le cadrage des exigences attendues de chaque aspect du système de gestion ferroviaire et leur suivi au cours des raffinements afin de garantir la couverture totale des besoins et des propriétés fondamentales du système.

L'utilisation de Pro-B pour réaliser la vérification des modèles fut également l'occasion pour moi d'utiliser des outils de développement formel pour un objectif plus mature que mes travaux habituels en entreprise, bien que des outils comme WP ou BinSec se révelent agréable à utiliser en comparaison de Rodin.

La preuve des modèles s'est révélée plus complexe que prévu, notamment en raison de la quantité et de la complexité des obligations de preuves générées par Rodin, par rapport aux preuves vues lors de ma formation.

5.3. REDACTED n°3

Notre projet a commencé la définition des tâches de chacun. Nous nous sommes répartis au fur et à mesure suivant ce que nous avions à faire. Pour commencer, nous avions défini qu'il fallait faire la modélisation ainsi que les exigences. De mon côté, je me suis lancé dans la modélisation, avec pour but de la représenter de manière graphique. C'est ainsi que j'ai un affichage similaire à celui présenté dans le sujet pour la modélisation. Evidemment, la modélisation a été modifiée tout au long du projet, et cette base de représentation a pu persister afin de mieux nous aiguiller lorsque nous en avions besoin. Lorsque j'ai fini de modéliser la première partie, le code a pu être débuté. J'ai alors continué la modélisation des parties suivantes. Lorsque cela fut fait, je me suis mis à faire le rapport.

Ceci a permis d'avoir dès le début un template que l'on pourrait alimenter tout au long du projet. Lorsque le raffinage 0 de la première partie eût été fait, j'ai fait le raffinage 1 et 2. Bien que fonctionnelles, ces parties ont été modifiées afin d'être plus en accord avec les exigences.

J'estime que j'ai eu de bonnes initiatives dès le départ afin de poser certaines bases qui nous ont aidé par la suite. Je pense que j'aurais pu faire plus attention à certaines exigences lors du code, bien que je sois tout de même content de ce que j'ai produit qui a permis d'aider tout le groupe.

Enfin, bien que j'ai apprécié le faire, j'estime que le sujet est trop conséquent par rapport à la période d'examen. Ceci a réduit le plaisir que j'ai pu prendre au départ, en laissant place à la nécessité de le faire pour une date de rendu.

6. Conclusion

Ce projet nous a permis de passer de la partie théorique vue en cours, à une partie plus pratique. Le projet présentant des aspects plus complexes que les Travaux Pratiques, il était en effet divertissant de s'accommoder des exigences définies parfois avec ambiguïté et d'être confronté à un seul document de référence. Le raffinement était libre mais assez bien guidé pour permettre un découpage assez logique et pratique à réaliser. La complexité du sujet et de sa modélisation nous a aussi souvent poussé à remettre en cause certaines hypothèses ou représentations qui méritaient d'être précisées. Néanmoins, nous avons été assez désarmés face à la complexité des preuves à réaliser. Avec notre niveau actuel dans ce domaine, nous ne savons pas s'il s'agit d'un problème de modélisation ou bien d'un manque de compétence dans l'écriture des preuves de notre part. La complexité du sujet nous a aussi contraint à ne pas pouvoir développer la dernière partie du sujet de part l'incomplétude / inexactitude de notre modèle vis-à-vis des exigences. Nous avons tout de même, malgré ces quelques inconvénients, trouvé le projet intéressant et mettant suffisamment à l'épreuve les compétences que nous avons pu développer dans cette matière Développement Formel de Systèmes.