Système de Gestion Ferroviaire

Projet de Développement Formel de Système
3A-SIL – 2024-2025

Les systèmes ferroviaires et les méthodes formelles ont une longue histoire en commun. Que ce soit les métros autonomes à commande numérique (comme la ligne 14 du métro de Paris) ou plus généralement les systèmes de signalisation et de contrôle-commande des trains (par exemple dans le standard ETCS [1]), on retrouve typiquement les méthodes formelles, et en particulier la méthode B/Event-B, à diverses étapes de conception.

Ce projet a pour but de modéliser un système ferroviaire, qui se compose de trains à router sur un réseau ferré, et qui sont chacun des objets physiques en mouvement. On utilisera pour cela la méthode Event-B, et exploiterons l'opération du raffinement, afin de simplifier le développement et la preuve.

Objectifs pédagogiques :

- lire une spécification informelle, en extraire une spécification formelle
- identifier les types d'exigences et leur représentation dans un formalisme
- concevoir une chaîne de raffinement, un plan de développement
- modéliser un système en respectant des exigences, faire le lien entre les éléments du système et les éléments de la spécification formelle
- utiliser Event-B (et la plate-forme Rodin) pour réaliser les modèles, décharger les obligations de preuve, faire des animations/simulation
- synthétiser, justifier, argumenter les techniques et les choix de développement utilisés (dans un rapport)

1 Cahier des charges

On répartit la spécification en plusieurs « couches logiques ». Ces couches sont des ensembles d'exigences propre à une partie du système seulement, et doivent s'articuler pour former un système complet.

1.1 Couche « Topologie et Signalisation »

Dans cette couche, on s'intéresse principalement au *réseau ferré* et à la place générale des trains. La partie signalisation du système gère les *routes* des trains, et garantit notamment que les trains ne se croisent jamais.

Réseau ferré Un *réseau ferré* est un ensemble connexe de voies. Les voies peuvent présenter des séparations et des regroupements (au travers d'aiguillages), autrement dit le réseau peut présenter des ramifications. Pour une exécution donnée du système, le réseau est fixe.

1. `https://en.wikipedia.org/wiki/European_Train_Control_System`

Tronçons Une voie est découpée en *tronçons*. Le découpage est arbitraire et fixe, et on suppose que les tronçons qu'occupent le train sont toujours suffisamment grand pour l'accueillir dans sa totalité.

Arrêt Un *arrêt* est un point particulier du réseau, connecté à exactement un tronçon d'entrée et un tronçon de sortie. Il peut s'agir du quai d'une gare, d'un hangar dans un centre de tri, d'une voie de déchargement, etc.

Trains Un *train* est un objet physique géré par le système, en circulation sur le réseau ou à un arrêt. Pour la signalisation, un *train* est principalement caractérisé par le(s) tronçon(s) ou l'arrêt auquel il se trouve.

Pour éviter au maximum les collisions, le système impose qu'il ne peut y avoir **qu'au plus un train dans un tronçon**. Si le train doit changer de tronçon et que le tronçon suivant est occupé, il doit attendre que ce dernier se libère.

À noter qu'un train *peut occuper deux tronçons* : c'est le cas lorsqu'il sort du tronçon actuel et entre dans le tronçon suivant. Les deux tronçons en question *doivent alors être adjacents*.

Routes Un train est affecté à une *destination*, qui est un arrêt sur le réseau. Cet arrêt est connu au moment où le train quitte son arrêt actuel. Pour joindre cet arrêt, le système de signalisation affecte une *route* au train, qui est une suite de tronçons à parcourir. Cette route peut être incomplète (et ne pas mener à la destination) ; dans ce cas, si le train arrive au bout de sa route, il doit attendre que le système lui donne une nouvelle route. Le système peut aussi changer la route du train alors qu'il ne l'a pas terminée. Dans les deux cas, lorsque le train reçoit une nouvelle route, le premier tronçon est celui où le train se trouve (pour garantir une forme de continuité des chemins).

À noter que, comme un arrêt est connecté à exactement un tronçon d'entrée, un chemin qui atteint l'arrêt de destination a pour dernier élément le tronçon d'entrée de l'arrêt. On suppose que si le train entre dans ce tronçon, il finit par en sortir et par occuper l'arrêt associé.

1.2 Couche « Matériel Roulant »

Dans cette couche, on s'intéresse plus particulièrement au *matériel roulant*, autrement dit aux trains eux-mêmes, indépendamment des autres. À noter qu'ici on s'intéresse aux trains d'un point de vue « logique » principalement. Dans un second temps, nous introduirons des notions de systèmes physiques.

Trains Un *wagon* est un élément insécable d'un train. Un *train* est une succession de wagons qui se touchent, et lorsqu'un train se déplace, il en va de même pour tous ses wagons, de manière simultanée et uniforme. On appelle *taille* le nombre de wagons du train.

Le wagon de tête d'un train est généralement une voiture qui tracte les autres wagons, et que l'on appelle *motrice* (en réalité, un train peut présenter plusieurs motrices, actives en même temps ou non, mais cela n'est pas important pour cette modélisation, et sera donc ignoré).

Lien avec les tronçons Un train occupe un tronçon dès lors qu'au moins un wagon se trouve dans ledit tronçon. Lorsque le train franchit la limite entre deux tronçons, chacun de ses wagons passent du tronçon actuel au tronçon suivant, dans l'ordre. Le mécanisme est le même pour les arrêts.

Nous rappelons que par hypothèse les tronçons parcourus par un train sont toujours suffisamment grands pour accueillir ce train dans sa totalité.

1.3 Couche « Physique et Dynamique »

Dans cette couche, on s'intéresse précisément au comportement des trains en tant qu'*objets physiques*. Cette partie de la spécification a pour but de « simuler » l'évolution des trains au fil du temps.

Trains et position Un train est caractérisé par la position de sa tête (le bout de sa motrice) et celle de sa queue (le bout du dernier wagon). La *longueur* d'un train est la différence entre ces deux positions. Un train occupe toujours tout l'espace de sa longueur.

On supposera que l'*élasticité* du train est négligeable, et donc que sa longueur ne varie pas.

Autorité de mouvement On nomme *autorité de mouvement* (*MA* pour *movement authority*) la plage de positions sur laquelle le train a le droit de se déplacer. Pour simplifier, cette autorité de mouvement correspond généralement au minimum à la zone entre le train et la fin du tronçon actuel, mais peut être plus grande (si le tronçon suivant est libre, typiquement). On appelle *fin* de l'autorité de mouvement (*EoA* pour *end of authority*) la position maximale de l'autorité de mouvement, autrement dit *la position maximale que le train a le droit d'atteindre*.

L'autorité de mouvement est mise à jour au fur et à mesure que le train avance. En particulier, si le tronçon suivant est libre et que le train approche de la fin du tronçon actuelle, l'autorité de mouvement est renouvelée, pour permettre au train de changer de tronçon.

Un train n'a pas le droit de dépasser la fin de son autorité de mouvement. Il doit typiquement s'arrêter avant, si nécessaire.

Vitesse et accélération Un train est caractérisé par sa vitesse et son accélération (commune aux deux positions, naturellement). On rappelle que la variation de position sur un intervalle de temps donné est égale à la vitesse, et que la variation de vitesse sur un intervalle de temps donné est égale à l'accélération, ce que l'on écrit généralement (avec a accélération, v vitesse et p position) :

$$\frac{\mathrm{d}v}{\mathrm{d}t} = a, \ \frac{\mathrm{d}p}{\mathrm{d}t} = v \tag{1}$$

Pour simplifier, on supposera que le système est régi par un temps *discret*. Autrement dit, le temps progresse par « sauts » de Δt, Δt fixe.

À noter qu'un train **ne peut jamais reculer**. Autrement dit, sa vitesse est toujours positive. Dans l'éventualité où la vitesse décroît du fait d'une accélération négative, si elle atteint 0, alors elle reste à 0 et l'accélération devient nulle (on ne se met pas à reculer quand on freine...).

La vitesse d'un train est par ailleurs *bornée* par une constante V_{max} (de par les règles physiques qui régissent le comportement du train). De même, l'accélération maximum du train est bornée par une constante A_{max}.

Distance de freinage La distance nécessaire à un train en mouvement à une vitesse v donnée pour atteindre une vitesse nulle ($v = 0$) avec la décélération/puissance de freinage donnée est appelée *distance de freinage* (ou SD pour *stopping distance*). C'est une quantité difficile à évaluer,

on supposera qu'il existe un algorithme qui la détermine avec précision, sans se soucier des détails d'un tel algorithme.

On suppose que le train a une *capacité de freinage* limitée, autrement dit une valeur *négative* minimum pour l'accélération ($F_{max} < 0$).

Modes et dynamique Un train contrôle son accélération directement. Un train est associé à un *mode*, un état qui dicte la façon dont il évolue. On distingue les modes suivants :

- *Mouvement libre* : le train avance librement sur le réseau, l'accélération a une valeur quelconque ;
- *Freinage* : le train freine, son accélération est négative et la vitesse est donc décroissante ;
- *Attente* : le train est immobile, sa vitesse et son accélération sont nulles ;

Changement de mode Les changements de modes sont régis par l'état du matériel roulant (lien avec la couche précédente) :

- Un train en mode *Mouvement libre* passe en mode *Freinage* dès que sa distance de freinage coïncide avec son autorité de mouvement ;
- Un train en mode *Freinage* passe en mode *Attente* dès que sa vitesse atteint 0 ;
- Un train en mode *Freinage* peut repasser en mode *Mouvement libre* si sa distance de freinage est plus petite que son autorité de mouvement ;
- Un train en mode *Attente* peut passer en mode *Mouvement libre* si son autorité de mouvement le lui permet (qu'elle vient d'être renouvelée, par exemple) ;

1.4 Couche « Contrôle » (Optionnelle)

Une modélisation plus fine de la couche précédente est possible, qui est davantage en adéquation avec la réalité. Dans cette couche, nous précisons le comportement physique du matériel roulant en y incluant des principes d'inertie. Nous ajoutons également un aspect de vitesse limite sur les tronçons.

Force motrice/de freinage Le train ne contrôle pas directement son accélération, mais plutôt la résultante de la sortie de son moteur et de l'action de ses freins, que l'on note f. Lorsque f est positif, on parle de force *motrice* (qui pousse le train en avant) et lorsqu'il est négatif, il s'agit d'une force *de freinage*. Dans la suite, on utilisera le terme *force motrice* indifféremment pour ces deux aspects.

Le lien entre la force motrice et l'accélération est donné par l'équation de Davis :

$$a = \frac{\mathrm{d}v}{\mathrm{d}t} = f - (A + Bv + Cv^2) \tag{2}$$

Où A, B et C sont des coefficients positifs fixes et connus, qui dépendent (entres autres) de la topologie du réseau et des mensurations du train [2]

À noter que f est borné : il existe une force motrice maximale ($f_{max} > 0$) et une force de freinage maximale ($f_{min} < 0$), et on a donc $f_{min} \leq f \leq f_{max}$.

2. À titre d'exemple et pour votre curiosité personnelle, les TGV en France ont pour coefficients de Davis $A = 25$ $\mathrm{m\,s^{-2}}$, $B = 1.188$ $\mathrm{s^{-1}}$ et $C = 7.03729\mathrm{e}^{-2}$ $\mathrm{m^{-1}\,s^2}$; avec une force motrice maximale f_{max} de 50 $\mathrm{m\,s^{-2}}$.

Vitesse limite par tronçon Chaque tronçon est associé à une unique vitesse limite, connue et fixée à l'avance. Sur toute la longueur du tronçon, le train n'a pas le droit de dépasser cette vitesse. Lorsqu'un train approche un tronçon dont la vitesse limite est inférieure à sa vitesse actuelle, il doit prendre toutes les mesures nécessaires pour atteindre cette nouvelle vitesse *avant* d'entrer dans le tronçon. Inversement, si la vitesse du tronçon suivant est *supérieure*, il n'a le droit de commencer à accélérer qu'une fois complètement dans le tronçon.

Le train a bien sûr tout à fait le droit d'avoir une vitesse inférieure à la vitesse limite. Par ailleurs, tant que sa vitesse est inférieure, il a le droit d'accélérer librement.

1.5 Indications

Pour vous mettre sur la bonne voie, nous vous donnons quelques indications. Il s'agit généralement de suggestions, libre à vous de les appliquer ou non, en fonction de vos mdèles.

Modélisation du réseau Pour représenter le réseau, nous vous conseillons la formalisation suivante :

— un type/ensemble de travail $TRONCONS$[3] pour représenter les tronçons du réseau ;

— une constante qui représente le réseau, qui est un graphe de tronçons, donc une relation $Reseau \in TRONCONS \leftrightarrow TRONCONS$.

Avec cette formulation, pour deux tronçons $t_1, t_2 \in TRONCONS$, t_1 et t_2 sont *connexes* (ou « adjacents ») si et seulement si $t_1 \mapsto t_2 \in Reseau$. L'ensemble des tronçons adjacents à t_1 est alors donné par l'image relationnelle $Reseau[\{t_1\}]$.

À noter que dans cette formulation, aucune hypothèse n'est faite sur le réseau ; libre à vous de rajouter des hypothèses **si vous le jugez pertinent**. Par exemple, la relation n'est pas *symétrique*, ce qui signifie que deux tronçons peuvent être adjacents dans un sens mais pas nécessairement dans l'autre (mais ce n'est peut-être pas un problème, au contraire...). Autre propriété absente : un tronçon peut être relié à lui-même (relation non-irréflexive), ce qui peut créer des incohérences (peut-être).

```
CONTEXT Ctx_XXX_Instance
EXTENDS Ctx_XXX
CONSTANTS t1, t2, t3, t4
AXIOMS
 -- Pour l'animation : 4 troncons seulement
 axm1: partition(TRONCONS, {t1}, {t2}, {t3}, {t4})
 -- Réseau simpliste
 axm2: Reseau = { t1 ↦ t2, t1 ↦ t3, t2 ↦ t1, t2 ↦ t4, t3 ↦ t4, t4 ↦ t3 }
END
```

FIGURE 1 – Exemple de contexte d'instanciation pour Pro-B, qui donne des valeurs pour le réseau

Utilisation de ProB Lorsqu'on utilise ProB pour l'animation, ce dernier choisi une réalisation des types et des constantes utilisés qui est globalement assez aléatoire, et généralement minimale.

Pour pallier cela, vous pouvez créer des contextes à part, qui étendent le contexte de la machine à animer, et qui donnent des valeurs précises aux types et constantes utilisées, pour forcer ProB à

3. Les noms sont des suggestions, libre à vous de les changer (dans la limite du raisonnable) pour les adapter à votre style

avoir une instance fixe lors de l'animation (voir par ex. Figure 1). Il vous suffit alors de remplacer le contexte vu par la machine avec ce nouveau contexte d'instance lorsque vous voudrez l'animer (n'oubliez pas de remettre le bon contexte avant de faire les preuves !).

Ce genre de contexte de paramétrage est très utile pour retirer la partie du non-déterminisme qui ne vous intéresse pas. Bien sûr, il faut faire attention à ne pas introduire d'inconsistances (définition incompatible avec les axiomes du contexte), sous peine d'obtenir des résultats erronés.

2 Consignes et livrables

L'objectif du projet est de produire un *modèle Event-B* qui réalise la spécification présentée ci-dessus. En particulier, le modèle obtenu doit respecter les propriétés (sûreté, vivacité) identifiées dans la spécification (par ex. les trains ne rentrent jamais en collision).

S'agissant de modélisation formelle, chaque composant du modèle *doit être prouvé* intégralement et, dans la mesure du possible, validé avec ProB.

La réalisation du modèle doit vous conduire à reformuler le cahier des charges de manière formelle, et nous attendons que vous explicitiez les *exigences formelles* du systèmes, et que vous fassiez le lien explicite entre ces exigences et les éléments de votre modèle.

2.1 Méthodologie

Le cahier des charges fourni est volontairement informel, ambigu et incomplet. Un objectif du projet est d'extraire, de ce cahier des charges, des *exigences* précises et « atomiques » (c'est-à-dire portant sur un minimum de choses à la fois), qui correspondent à des éléments du modèle proposé (ensembles/constantes, axiomes/théorèmes, variables, invariants, événements, etc.).

Nous recommandons de numéroter les exigences extraites, afin de pouvoir y faire référence et les relier à des éléments du modèle.

Il est aussi intéressant d'identifier en particulier les exigences relatives à la *sûreté* du système, et celles qui sont des *hypothèses* du système (propriétés admises de certains éléments, règles physiques, etc.). On pourra, par exemple, identifier les exigences de sûreté avec le code **SAF**, les hypothèses avec le code **HYP**, et les autres exigences avec le code **REQ** (pour *requirement*).

REQ-1	Les trains se déplacent sur le réseau
HYP-1	Le réseau se compose de tronçons connexes
SAF-1	Il ne doit pas y avoir deux trains dans un même tronçon

TABLE 1 – Exemple d'exigences extraites, catégorisées et numérotées

Les exigences considérées varieront nécessairement au fur et à mesure de la conception du système. Typiquement, chaque machine proposée répondra à *certaines* exigences. Il est également possible que vous formuliez des exigences dont vous avez besoin (pour de la preuve, typiquement) ou qui vous semblent pertinentes mais ne sont pas directement mentionnées dans le cahier des charges ; c'est normal et attendu, la spécification étant ambiguë. Vous devrez identifier ces exigences et les démarquer (avec un symbole ou une couleur particuliers, par exemple).

Nous insistons sur le fait qu'il est normal de revenir sur des exigences au cours de la modélisation. Il est habituel que l'on doive détailler ou reformuler des exigences à mesure que le modèle se précise.

2.2 Modélisation

Vous devrez concevoir un *modèle Event-B*, qui se compose de *machines* reliées par une relation de raffinement, et d'éventuels contextes regroupant les éléments nécessaires à la modélisation.

Il vous appartient de concevoir une chaîne de raffinement, de décider comment s'organise chaque étape, ce que contient chaque contexte et machine, etc. Néanmoins, il nous semble pertinent de vous intéresser à chaque couche l'une à la suite de l'autre, avec chaque couche formée elle-même de plusieurs raffinement.

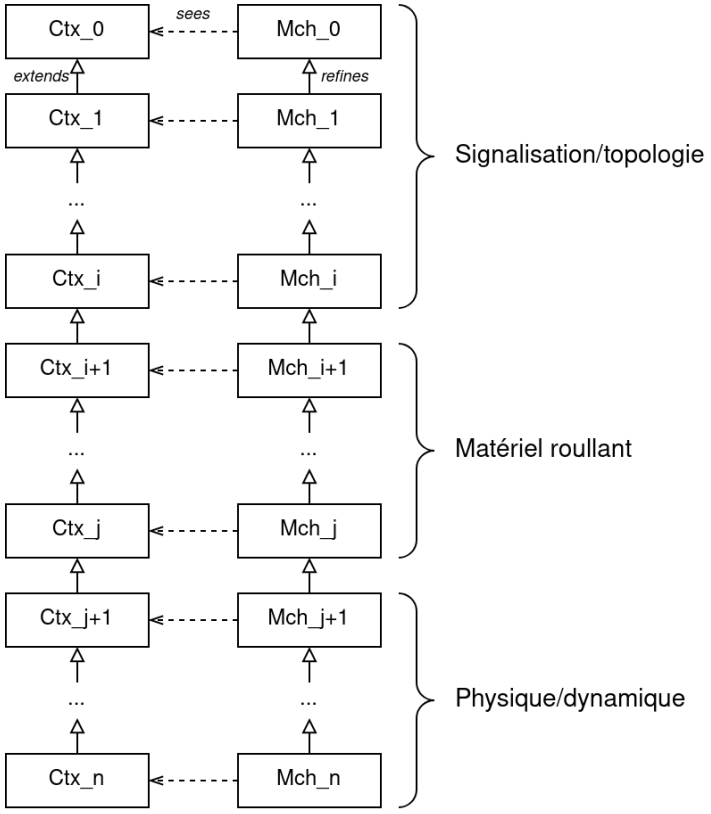

FIGURE 2 – Schéma général recommandé (noms et nombre de machines arbitraires)

Chaque composant (contexte ou machine) encode des exigences, qu'il vous faudra identifier clairement.

À noter qu'un schéma de raffinement n'est jamais définitif. Il ne vous est pas interdit de revenir dessus, d'ajouter des étapes, de changer des machines au milieu de la chaîne, etc. Il est rare de réussir à avoir un plan correct du premier coup !

L'écriture des modèles doit être faite avec la plus grande rigueur : noms pertinents et respectant les conventions, exactitude des formules (invariants, gardes, etc.) et utilisation de commentaires.

Les modèles doivent être prouvés au maximum. À noter qu'en général, on prouve un modèle avant de la raffiner, la preuve pouvant révéler des problèmes dans le modèle (la validation avec ProB est également un bon moyen d'identifier les manquements du modèle).

2.3 Documentation technique

Le développement du modèle s'accompagne de l'écriture d'un rapport technique, dont le but est d'expliquer en détail les étapes de la réalisation du projet.

Exigences et modèles Le rapport doit impérativement présenter les exigences extraites du cahier des charges, et leurs liens avec les éléments du modèle. Bien sûr, cela suppose de présenter les modèles eux-mêmes.

Nous vous conseillons de restreindre la présence de gros morceaux de code au minimum, et de privilégier des petits extraits, et seulement si vous jugez que c'est nécessaire. L'explication des modèles passera donc nécessairement par des descriptions formelles et informelles, des diagrammes, etc.

Validation et vérification Le rapport doit présenter les activités de validation (model-check) entreprises durant la réalisation du projet : animation avec ProB, validation d'invariants et de raffinement, écriture de propriétés LTL, et ainsi de suite (des captures d'écran de ProB peuvent être utiles pour appuyer votre discours).

L'activité de preuve doit également être abordée. Inutile d'inclure *in extenso* des arbres de preuve complets, mais il est intéressant de donner quelques statistiques : nombre d'obligations de preuve générées (éventuellement séparées par type) et nombre d'obligations de preuve automatiques. Vous pouvez également mentionner les preuves que vous trouvez particulièrement intéressantes, celles qui vous ont donné du fil à retordre voire qui vous on conduites à revenir sur votre modélisation...

Note : il est possible d'obtenir des statistiques sur les obligations de preuve à l'aide de la vue *Statistics* (accessible depuis le menu *Window* > *Show view*).

Bilan personnel La conclusion du rapport doit contenir un *bilan personnel* (un pour chaque membre du groupe de projet) d'une dizaine à une quinzaine de lignes, qui présente le rôle du membre, une auto-critique du travail réalisé (points positifs, points négatifs, points manquants), et enfin une appréciation générale du projet (intérêt, difficulté, etc.).

Structure Le rapport doit présenter une structure claire, pertinente et efficace, détaillée sous forme d'une table des matières au début du document. Il doit impérativement contenir une introduction, qui présente le sujet *avec vos propres mots*, et terminer avec une conclusion qui fait un bilan des actions réalisées, de ce qu'il resterait à faire (le cas échéant), des avantages et défauts de la modélisation.

L'inclusion de figures (diagrammes, captures d'écran, etc.) est plus que vivement recommandée pour appuyer votre discours.

Typographie et contraintes éditoriales Le plus grand soin doit être apporté à l'écriture du rapport. Il doit être rédigé en français, dans une police d'écriture homogène avec empâtement (Times, Garamond, etc.) et en taille 11 points, sur papier A4. Les règles éditoriales françaises doivent être respectées (retrait pour chaque paragraphe, titres numérotés, etc.).

La grammaire et l'orthographe doivent être soignés, et les phrases doivent être cohérentes, structurées et compréhensibles. Il est impératif d'utiliser, de manière adéquate et précise, les éléments de jargon et de vocabulaire scientifique abordé dans cette UE.

Enfin, le rapport doit présenter une page de garde, comportant au minimum un titre, le nom de l'UE, l'année universitaire et la liste des membres du groupe, par ordre alphabétique du nom de famille.

L'usage d'un modèle de langage et assimilés (type GPT) pour générer tout ou partie du rapport est strictement interdit dans le cadre de ce projet, et sera considéré comme de la triche.

Conformément à l'article L122-5 n° 3 du code la propriété intellectuelle, il est possible d'utiliser une oeuvre rendue publique pour contribuer à son discours à condition d'en indiquer clairement et sans ambiguïté la provenance et l'auteur, et de clairement identifier la citation en tant que tel.

2.4 Modalités de rendu et d'évaluation

Le projet se réalise en groupe de 4 (préférentiellement) ou 3, avec un seul rendu pour tout le groupe. Le rendu se compose de deux livrables :

1. Une archive contenant l'intégralité des modèles développés dans le cadre du projet ;
2. Le rapport technique au format PDF (pas de limite de page) ;

Nous donnons ici quelques critères d'évaluation à titre indicatif :

- Modèles :
  - Nommage et conventions : les éléments sont nommés en respectant les conventions et de manière pertinente ;
  - Correction : les modèles proposés sont corrects ;
  - Raffinement : la chaîne de raffinement proposée est pertinente, les raffinements proposés sont corrects ;
  - Pertinence : les modèles proposés représentent des aspects précis de la spécification, et les aspects relatifs au hypothèses du système, et à la sûreté, vivacité et autres propriétés sont correctement encodés dans les composants adéquats ;
  - Preuves, animation : les preuves sur les modèles ont été réalisées, et les modèles ont été animés et validés avec ProB (dans la mesure du possible) ;
  - Couverture fonctionnelle : les modèles réalisent au maximum les exigences du cahier des charges ;
- Rapport :
  - Structure : le rapport est correctement structuré, et respecte les contraintes données dans le sujet (introduction, conclusion, table des matières, page de garde) ;
  - Complétude : le rapport aborde tous les points demandés (notamment le bilan personnel), et tout ce qui est nécessaire à la compréhension du rendu ;
  - Pertinence : le rapport ne parle *que* de ce qui est nécessaire, et ne se perd pas dans des détails inutiles ;
  - Typographie, forme : le rapport est correctement mis en forme (suivant les exigences données dans la Section 2.3) et présente une orthographe et une grammaire soignés ;

Le rendu se fera par Moodle. Les dates buttoir vous seront communiqués ultérieurement.