Développement formel de systèmes complexes
3 SN - L
Projet 2021 - 2022

Système auto-adaptatif :
Automatic Rover Protection (ARP)

Dans le cadre de ce projet, nous proposons de modéliser un Système auto-adaptatif : L'Automatic Rover Protection (ARP).

1 Description informelle

L'objectif de ce projet est de produire, en utilisant le raffinement, un modèle décrivant la gestion de systèmes auto-adaptatifs. Le modèle résultant de ce développement permettra de supporter et de traiter différentes fonctionnalités des rovers.

L'Automatic Rover Protection (ARP) est une fonction d'évitement de collision pour un rover plongé dans un environnement comprenant d'autres rovers de même type. L'ARP est un système autonome comprenant un ensemble de rovers. Chaque rover est supervisé par une station de supervision unique pour tous les rovers.

L'objectif principal de l'ARP est d'assurer la sécurité des mouvements entre les rovers autonomes en utilisant un ensemble de contraintes dans le but d'éviter les collisions[1].

Chaque rover

- se déplace de manière autonome sur une voie prédéfinie en fonction de contraintes de sécurité données, en appliquant la vitesse et le freinage requis ;
- a une vue partielle de l'état des autres rovers i.e. une connaissance de l'état des rovers voisins qui peuvent avoir un impact sur les décisions prises par ce rover.

Le poste de supervision unique

- supervise périodiquement (donc régulièrement) chacune des actions (déplacements) de chaque rover, en particulier la position et la vitesse de chaque rover ;
- maintient également une information globale sur la totalité du système i.e. maintient un état global de l'ensemble des rovers. Cet état global est visible de tous les rovers et donc peut être utilisé par chaque rover en vue de définir des évolutions/actions futures ;
- peut également passer outre toute tâche ou action pouvant être effectuée par n'importe quel rover. Par exemple, le superviseur peut actionner un arrêt ou un freinage sur n'importe quel rover en cas d'urgence ou de difficultés techniques identifiées.

[1] Cette étude de cas correspond à plusieurs systèmes complexes où des entités autonomes cohabitent. Un cas particulier est le contrôle des avions au sol dans un aéroport.

2 Définitions

Pour les besoins de la description des différentes exigences, nous introduisons les définitions suivantes.

Chaque rover est associé à une zone

- **DEF-D1: Warning Area (Zone d'alerte - WA).** Zone dans laquelle un rover effectue une action d'urgence en présence d'un autre rover.
- **DEF-D2: Caution Area (Zone de précaution - CA).** Plus petite zone dans laquelle un rover peut s'arrêter sans atteindre la limite de cette zone.
- **DEF-D3: Info Area (Zone d'information - IA).** Zone dans laquelle un rover voit d'autres rovers sans devoir effectuer d'action.

En fonction de la vitesse et de la position, ces zones peuvent être caractérisées dans l'espace et dans le temps.

3 Exigences

Nous distinguons les exigences de sûreté (préfixées par **SAF**) et les exigences fonctionnelles (préfixées par **FUN**).

3.1 Exigences de sûreté

Les exigences de sûreté suivantes sont définies.

- **SAF-R1: Séparer les rovers dans le temps et l'espace.** Tous les rovers actifs doivent être séparés dans les domaines temporel et spatial. Si deux rovers ne sont pas séparés, on parle alors de "conflit" ;
  - ***SAF-R1.1: Séparer les rovers dans le domaine temporel.*** L'ARP doit détecter toute situation de conflit temporel et réagir instantanément lorsque deux rovers actifs deviennent temporairement trop proches.
  - ***SAF-R1.2: Séparer les rovers dans le domaine spatial.*** L'ARP doit détecter toute situation de conflit spatial et réagir instantanément lorsque deux rovers actifs deviennent trop proches dans l'espace.

3.2 Exigences fonctionnelles

Les exigences fonctionnelles suivantes sont définies.

- **FUN-R1. Etats d'un rover.** Un rover actif peut soit se déplacer en avant *"MOVE"*, soit être arrêté *"STOP"* ou bien en cours de freinage *"BRAKE"*.
- **FUN-R2. Détecter un conflit.** Un rover actif doit détecter toute situation de conflit dans un délai de 100 ms (au maximum). Le conflit détecté doit persister jusqu'à sa disparition.
- **FUN-R3. Fournir les données sur la position d'un rover.** Chaque rover doit communiquer sa position, son identifiant et la zone d'alerte délimitée requise à tous les autres rovers, ainsi qu'au superviseur.

- **FUN-R4. Priorités entre rovers.** La priorité entre rovers est fournie/calculée par une fonction bijective f, qui associe, à chaque identifiant de rover, un entier représentant son niveau de priorité.

- **REQ-R5. Éviter les conflits.** En cas de détection d'une (ou de plusieurs) situation(s) de conflit, un rover doit (i) déterminer une action de résolution, et (ii) appliquer cette action de résolution.
  Cette exigences est décomposée en plusieurs exigences fonctionnelles comme suit.

  - ***FUN-R5.1. Déterminer l'action de résolution.*** L'action de résolution du conflit doit toujours être réalisée dans le but de satisfaire les exigences SAF-R1.1 et SAF-R1.2 définies ci-dessus.
  - ***FUN-R5.2: Appliquer l'action de résolution de conflits d'arrêt "STOP".*** Si l'action de résolution de conflit est "STOP", le rover doit appliquer la force de freinage maximale jusqu'à l'arrêt complet du rover.
  - ***FUN-R5.3. Appliquer l'action de résolution de conflits de freinage "BRAKE".*** Deux cas sont possibles lorsque l'action de résolution de conflits "BRAKE" est déclenchée sur un rover actif.

    1. Si un rover actif n'a pas la priorité sur les autres rovers, le freinage doit être appliqué sur ce rover pour s'assurer qu'il quitte les zones de précaution (WA) des autres rovers actifs.
    2. Si un rover actif a une plus forte priorité sur les autres rovers, alors il n'entreprend aucune action.

- **FUN-R6. Action à l'initiative du superviseur.** Un rover effectue toute action d'évitement de collision ordonnée par le superviseur : "STOP" et "BRAKE". Si une action "BRAKE" est déclenchée par le superviseur, une décélération définie par le superviseur est également appliquée pendant que l'action "BRAKE" est active (force de freinage).

- **FUN-R7. Résolution des cas d'actions conflictuelles.** Cette résolution concerne le cas où deux actions sont déclenchées, de manière concurrente, par un rover, ou si une action est déclenchée alors qu'une autre est en cours,

  1. L'action "STOP" est déclenchée par un rover de manière autonome par rapport à l'action "BRAKE", i.e. "STOP" préempte "BRAKE"
  2. Une action déclenchée par le superviseur préempte une action déclenchée par un rover de façon autonome.

- **FUN-R8. Données relatives aux rovers qui quittent l'ensemble des "rovers connus".** Lorsqu'un rover quitte l'ensemble des "rovers connus", sa dernière position doit demeurer connue/accessible de tous les autres rovers.

- **FUN-R9. Recouvrer une action "STOP".** Lorsqu'une action "STOP" est déclenchée par un rover, alors, pour reprendre sa mission, ce rover doit attendre le déclenchement d'une action du superviseur.

- **FUN-R10. Ajouts et suppression de Rovers.** Le superviseur centralisé peut ajouter ou supprimer un rover de l'ensemble des rovers actifs..

3.3 Hypothèses sur l'environnement dans lequel évoluent les rovers.

- **ENV-E1. Capacité de freinage.** Selon le principe de Newton, un rover est capable d'utiliser la force maximale de freinage pour s'arrêter sans glisser.
- **ENV-E2. Vitesse limitée.** Un rover ne doit pas dépasser la vitesse limite associée à la voie sur laquelle il se déplace. La fonction de contrôle du rover doit veiller à garantir le respect de cette limitation.
- **HYP-E3. Gestion des rovers "hors piste".** Lorsqu'un rover devient "hors-piste", il déclenche l'action "STOP". Un rover est hors-piste en cas de panne. C'est le superviseur qui ordonne la mise hors-piste d'un reover.
- **HYP-E4: Gestion des ensembles "Active rovers".** Tous les rovers actifs sont contrôlés par le superviseur. Par exemple, l'ajout/suppression de rovers actifs doit être effectués par le superviseur. Cette information doit être diffusée instantanément à tous les autres rovers actifs.
- **ENV-E5.** Les rovers actifs se déplacent toujours sur un plan horizontal.

4 Objectifs du projet

Nous nous proposons de concevoir un modèle Event-B associé à la fonction de l'ARP permettant la gestion de l'évolution de plusieurs rovers. Ce modèle devra être construit graduellement avec la prise en compte pas à pas des différentes exigences.

Le raffinement sera mis en œuvre pour définir la séquence de modèles de rover allant du modèle abstrait jusqu'au modèle concret. Le dernier modèle de cette séquence devra prendre en compte la totalité des exigences. Il s'agira alors du modèle terminal.

Une fois ce modèle terminal obtenu, on identifiera les actions associées au superviseur et les actions associées à un rover. De cette façon, les actions implantées coté rover et celles implantées coté contrôleur seront correctement séparées.

5 Une modélisation possible

Une stratégie de raffinement possible est décrite ci-dessous. Les différentes exigences pouvant être prises en compte par chaque modèle.

- Modèle initial. Modes de fonctionnement du contrôleur
- Premier raffinement. Ségrégation spatiale et position
- Deuxième raffinement. Quantités physiques de l'ARP
- Troisième raffinement. Introduction de la dynamique et du contrôle associé
- Quatrième raffinement. Séparation des domaines spatiaux et temporels
- Cinquième raffinement. Introduction du temps au travers d'une horloge
- Sixième raffinement. Détection des conflits
- Septième raffinement. Résolution des conflits

6 Remarques

- Attention, la stratégie de raffinement de la section précédente est proposée par les auteurs du sujet de projet. Elle n'est pas unique, il n'est pas indispensable de la suivre. Vous pouvez définir votre propre stratégie, et surtout y réfléchir longuement pour ne pas à avoir à re-développer vos modèles à chaque fois que vous aurez identifié un problème.
- Il est important de positionner les différentes exigences prises en compte à chaque niveau de raffinement.
- Les exigences précédentes peuvent être complétées si vous jugez qu'elles ne sont pas suffisamment précises.
- Des exigences de sûreté peuvent également être introduites partout où cela est nécessaire.

7 Travail demandé.

Il est demandé de réaliser un développement prenant en compte les exigences précédentes.

Si vous définissez votre propre stratégie de raffinement, il est demandé de l'expliciter comme cela a été fait dans la section 5.

Pour chacun des modèles produit, il est demandé de suivre les étapes suivantes.

1. Définir le modèle associé à chaque niveau de modélisation.
2. Dériver un système états-transitions correspondant à ce modèle.
3. Utiliser la plate-forme Rodin comme support de modélisation
4. Compléter le modèle par les propriétés de sûreté pertinentes sous forme d'invariant
5. Valider le modèle en utilisant l'animation à l'aide du model checker ProB
6. Vérifier l'absence de blocage sur ce modèle en utilisant ProB.
7. Utiliser la logique LTL ou CTL pour décrire des propriétés pertinentes
8. Modifier les préférences de ProB pour permettre l'utilisation de 5 éléments dans les ensembles définis puis animer le même modèle avec ces nouvelles préférences modifiées.

8 Exigences pour la réalisation du projet

1. Le projet est réalisé en **groupes de 3 élèves**. Chaque élève devra préciser **dans le rapport final quelle aura été sa contribution** au projet.
2. Le langage de modélisation utilisé est le langage Event-B sur la plate-forme Rodin
3. Le modèle devra être valide et fonctionner sur les machines Linux de l'N7.
4. L'ensemble des concepts du cours devront être mis en œuvre, en particulier :
    - Conception en utilisant la méthode des raffinages

- Justification des choix des types de données manipulés
- Conception de composants : **Context** et **Machine**.
- Utilisation de ProB pour valider/animer les modèles
- Utilisation des prouveurs de la plate-forme Rodin pour prouver les différentes obligations de preuve des modèles produits.

9 Livrables

Les **documents à rendre** sont :

- une archive contenant les modèles **Event-B**
- un **rapport** (rapport.pdf) contenant au moins :
  - un résumé qui décrit l'objectif et le contenu du rapport (10 lignes max),
  - une introduction qui présente le problème traité et le plan du document
  - l'architecture du développement
  - la justification des principaux choix réalisés
  - la présentation des principaux raffinement et types de données
  - la démarche adoptée pour animer les modèles avec ProB
  - les difficultés rencontrées et les solutions adoptées en justifiant vos choix (en particulier quand vous avez envisagé plusieurs solutions)
  - un bilan technique donnant un état d'avancement du projet et les perspectives d'amélioration/évolution
  - un bilan *personnel* et *individuel* : intérêt, temps passé, temps passé à la conception, temps passé à la modélisation, la preuve et la validation, temps passé à la mise au point, temps passé sur le rapport, enseignements tirés de ce projet, etc.