École Polytechnique Montréal

Département de Génie Informatique et Génie Logiciel

LOG8470

Vérification de la fiabilité et de la sécurité

Travail pratique 1

Model Checking - iSpin

Soumis par

Ibrahima Séga Sangaré (1788085)

Le 1er novembre 2018

Introduction

Ce TP porte sur la modélisation de systèmes avioniques comportant plusieurs modules ayant des fonctions de niveau de criticité variable. Concrètement, dans les systèmes avioniques modernes, on constate la présence d'architecture de type *IMA* (*Integrated Modular Architecture*). Cette architecture permet l'hébergement de plusieurs fonctions avec des niveaux de criticité différents sur une même plateforme. Ces fonctions sont contenues dans des modules qui sont interconnectés par un commutateur qui gère l'envoi des messages provenant des fonctions du systèmes.

Dans notre cas, un exemple de système avionique représentant a été étudié pour l'exercice de modélisation. Le système est divisé en trois modules M_1, M_2 et M_3. Le module M_1 contient la fonction FCS (*Fuel Control System*) et ECS (*Entertainment Control System*). Le module M_2 contient la fonction LGS (*Landing Gear System*). Finalement, le module M_3 contient la fonction MFD (*Multi Functional Display*). Ce module n'a pas été conservé pour le reste travail pratique.

Les objectifs de ce TP ont donc consister à modéliser le système avionique basée sur l'IMA et d'utiliser le logiciel iSPIN pour vérifier les propriétés de model checking.

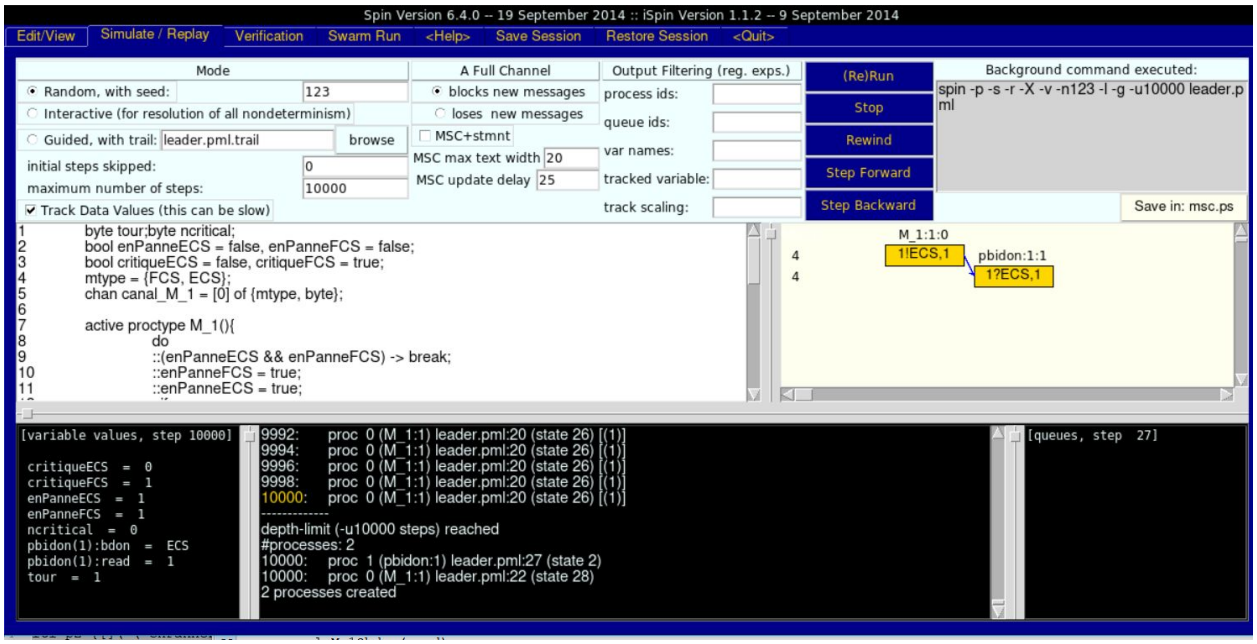
Présentation des Travaux

Partie 1

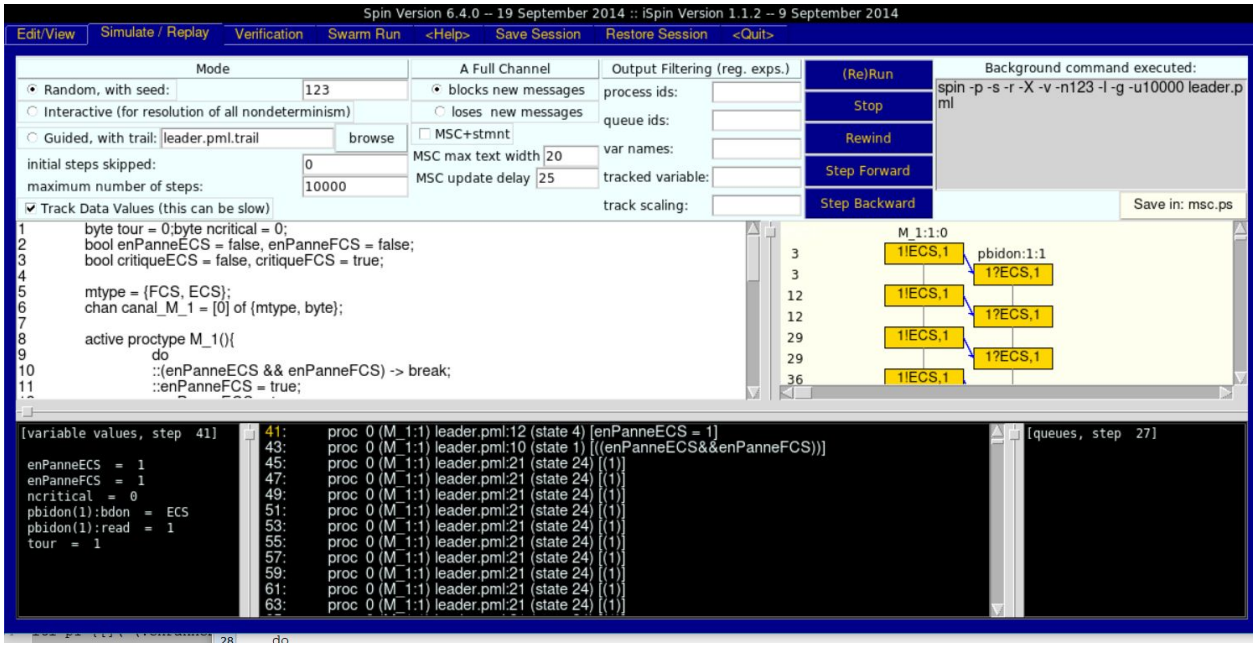
La première partie du TP, le module M_1 a été modélisé en PROMELA avec l'existence des deux fonctions FCS et ECS. Des contraintes ont été considérées pour cette modélisation. D'abord, les fonctions avaient la possibilité de tomber en panne à tout moment. Elles envoient chacune des signaux uniques et déposent des messages tant qu'elles sont actives. Le système doit être modélisé de telle sorte que, si l'une des fonction tombe en panne l'autre peut continuer à fonctionner. L'exclusion mutuelle doit être assurée également lorsqu'il y a un envoi de messages.

Pour la modélisation, une variable globale partagée à été utilisée pour assurer l'exclusion mutuelle. Chaque processus passe la main après l'envoi d'un message et peut tomber en panne en brisant le cycle de dépôt de messages à l'aide d'un *'break*'. Les trois propriétés demandées ont été implémentées en LTL et sont observables dans le code Promela. La première propriété (exclusion mutuelle) est une propriété de safety, la seconde est une propriété de liveness et la troisième est une propriété de liveness.

En vérifiant ces propriétés, un problème est apparu lors de la validation des deux dernières propriétés. En effet, le problème survient lorsque une des fonctions tombe en panne et ne passent pas la main au processus qui tombent en panne. Une trace d'exécution a été associée à ce problème.



Pour pallier à ce problème, le processus qui tombe en panne met le tour à jour pour éviter que le processus ne soit pas en attente de son tour et éviter que le système en entier ne tombe en panne.



Les questions 5.1 à 5.3 sont traités par le code ‘Part1.pml’.

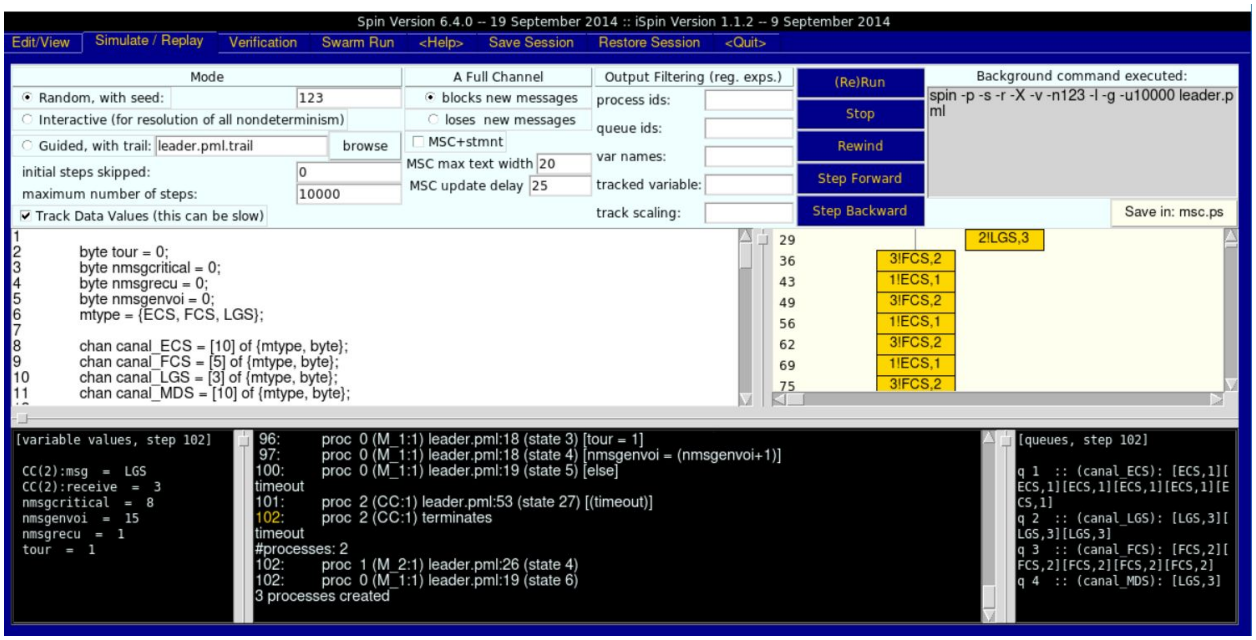
La question 5.4 est traitées par le code ‘Part1-Amélioration.pml’.

Partie 2

Q6.1 à Q6.3

Dans la deuxième partie du tp, le travail effectué à consister à modéliser une partie de la communication réseau dans le système avionique. La modélisation a consisté à la représentation en PROMELA des modules M_1 et M_2 qui communiquent avec le commutateur pour déposer différents types de messages dans un canal. Pour ce faire, les fonctions FCS et ECS déposent des messages à tour de rôle en se synchronisant avec le commutateur. La fonction LGS, quant à elle, dépose des messages sans restriction. La première propriété (a) est une propriété de liveness et la seconde (b) constitue une propriété de safety. Elles se retrouvent dans le code PROMELA.

Dans un premier temps, l'ordonnancement a été considéré de façon équitable entre les fonctions ECS et FCS lors de l'envoi de messages. Cela a été le cas également entre les deux modules. Cela a causé un problème par rapport à la deuxième propriété, car les messages de haute criticité arrivaient à se perdre à cause de défaut d'ordonnancement. Une trace d'exécution a été associée à ce problème.



Q6.4 à Q6.5

Pour pallier à ce problème, les envois des messages de haute criticité ont été placés en priorité dans le commutateur, le problème n'a pas été réglé car il y avait toujours la possibilité de que l'ordonnancement ne donne pas la propriété aux fonctions de haute criticité.

Bonus

En attribuant des priorités aux modules M_1 et M_2, le problème d'ordonnancement est réglé en partie car les messages critiques sont plus souvent considérés même si le problème n'est pas complètement effacé.

Un code promela est associé aux questions correspondantes.

Les questions 6.1 à 6.3 sont traitées par le code 'Part2.pml'.

La question 6.4 est traitée par le code 'Part2-Q4.pml'.

La question bonus est traitée par le code 'Part2-Bonus.pml'.

Difficultés Rencontrées

Parmi les difficultées rencontrées, il a été difficile dans la première partie de trouver une solution qui permettait de gérer les situations de panne de manière efficace et respecter, par la même occasion, les propriétés du système. En utilisant l'algorithme de Peterson et s'inspirant de la documentation de SPIN, il a été possible de générer une solution qui respecte les consignes.

Dans la deuxième partie, il a été difficile de représenter correctement la communication réseau, car il fallait s'assurer de garder les messages de haute criticité. En réutilisant , les concepts de synchronisation de la première partie, il a été possible de trouver une solution.

Conclusion

En conclusion, ce TP a permis une bonne introduction au langage PROMELA, à la modélisation de système et au model checking. La présence des spécifications LTL a également permis de comprendre davantage comment les modéliser et leur donner un sens concret. Le TP a été bien adapté pour un début. On peut s'attendre à ce que le prochain TP continue sur la même lancée.