

Génie Logiciel 2

M1/GL2

TD 3 : Modélisation avec MCRL2

Tarek Melliti & Pascal Poizat

2011-2012

Exercice - 1 *Préambule*

MCRL2 est un ensemble d'outils permettant de modéliser et de vérifier des systèmes décrits en termes de processus et/ou de systèmes de transitions étiquetés (LTS).

1- Si ce n'est pas déjà fait (c'est-à-dire, si vous ne venez pas en cours), installez MCRL2.

Pour cela, allez sur le site <http://www.mcrl2.org/>. Une documentation très riche (en plus des explications fournies en cours) est par ailleurs disponible sur ce site.

2- Si vous désirez utiliser les outils en ligne de commande, ajoutez le chemin vers eux dans votre fichier de démarrage (par exemple ici pour le fichier `$HOME/.bash_profile` sous MacOS) :

```
# MCRL2
export MCRL2=/Applications/mcrl2.app/Contents/
export PATH=$PATH:$MCRL2/MacOS
export PATH=$PATH:$MCRL2/MacOS/ltsview.app/Contents/MacOS
export PATH=$PATH:$MCRL2/MacOS/ltsgraph.app/Contents/MacOS
export PATH=$PATH:$MCRL2/MacOS/diagraphica.app/Contents/MacOS
export MANPATH=$MANPATH:$MCRL2/share/man
```

3- Vous pouvez aussi utiliser l'interface graphique fournie avec MCRL2.

Exercice - 2 *Algorithme de Peterson (Prise en main)*

On va vérifier l'algorithme de Peterson vu en cours (voir les transparents). Cet exercice est une prise en main et doit être fait rapidement sachant que nous l'avons fait en cours.

1- Saisissez le modèle en mcrl2 de l'algorithme dans un fichier `peterson.mcrl2`

2- Créez le LTS correspondant dans un fichier `peterson.lts` et visualisez le.

3- Vérifiez l'absence de blocage dans le système.

Exercice - 3 *Buffers (Raffinement et correspondance entre spécification et implantation)*

Dans cette première partie, vous ne prendrez pas en compte les données.

1- Spécifiez dans un fichier `b2_spec.mcrl2` le comportement d'un buffer à deux places, `B2_SPEC`.

2- Décrivez dans un fichier `b2_impl.mcrl2` le comportement d'un buffer à une place, `B1_IMPL`. Complétez en décrivant la mise en parallèle `B2_IMPL` de deux buffers à une place en synchronisant la sortie du premier avec l'entrée du deuxième.

3- Montrez que B2_IMPL est un raffinement (implantation correcte) de B2_SPEC.

Dans cette seconde partie, vous prendrez en compte les données (entiers [0..5]).

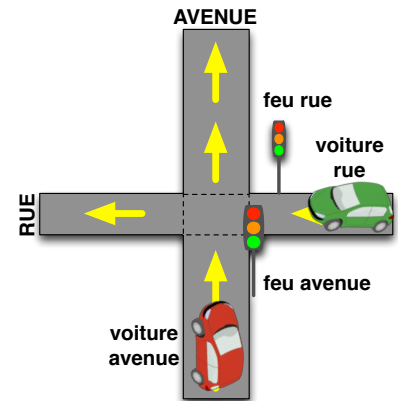
4- Refaire les questions 1 à 3 avec la prise en compte des données dans les processus et dans leurs échanges. Vous le ferez dans des fichiers de noms différents, par exemple b2d_spec.mcrl2 et b2d_impl.mcrl2.

Exercice - 4 Carrefour dangereux (Vérification avec observateur(s))

Soit le programme suivant modélisant un carrefour dans lequel les voitures contrôlent les feux.

```
feuAvenue := vert; feuRue := vert;
```

Processus VoitureAvenue:	Processus VoitureRue:
A: feuRue := rouge	A: feuAvenue := rouge
B: ATTENDRE (feuAvenue==vert)	B: ATTENDRE (feuRue==vert)
C: V.A. entrer croisement	C: V.R. entrer croisement
D: V.A. sortir croisement	D: V.R. sortir croisement
E: feuRue := vert	E: feuAvenue := vert



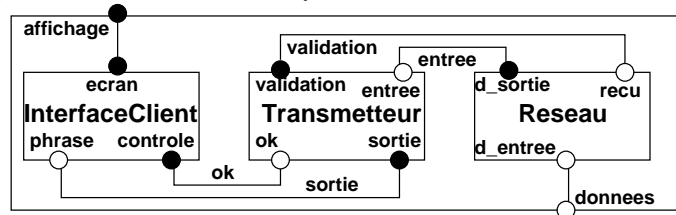
- 1- Modélisez ce système de carrefour (quatre processus en parallèle, 2 voitures et 2 feux)
- 2- Le système peut-il se bloquer ?
- 3- Le système assure-t-il la sécurité du carrefour (utilisez un processus observateur pour répondre) ?

Exercice - 5 Étude de cas

On désire modéliser un client de système de dialogue entre utilisateurs. Ce qui est attendu de ce système est de transmettre de façon répétitive les données reçues sur un affichage :

```
?donnees !affichage ?donnees !affichage ...
```

Le système a été implanté à l'aide de trois sous-systèmes communicants :



Le comportement des trois sous-systèmes est le suivant :

- la connexion réseau (Reseau) reçoit des données (d_entree) d'un contact extérieur puis les envoie au transmetteur (d_sortie). Elle reçoit ensuite un reçu (recu) de sa part puis peut recommencer à recevoir des données (et ainsi de suite).
- le sous-système de transmission (Transmetteur) est capable de recevoir des données de la connexion réseau (entree) puis de les transmettre (sortie) à l'interface graphique. Il peut aussi recevoir des messages de l'interface (ok) et les transmettre à la connexion réseau (validation). Chacune de ces deux capacités se fait de façon séquentielle. Il n'y a pas de limite au nombre de fois que le sous-système de transmission peut faire ces capacités.
- l'interface graphique (InterfaceClient) peut recevoir une phrase (phrase) et l'envoyer sur l'écran (ecran). Elle renvoie ensuite une information de contrôle pour le signaler (controle) et se remet en attente d'une phrase (qu'elle enverra sur l'écran et ainsi de suite).

- 1- Modélisez Reseau, Transmetteur et InterfaceClient
- 2- Donnez l'expression mcrl2 correspondant au diagramme architectural
- 3- Montrez que l'implantation est incorrecte. Corrigez.