GénéSyst : Génération d'un système de transitions étiquetées à partir d'une spécification B événementiel

Xavier Morselli, Marie-Laure Potet, Nicolas Stouls

GénéSyst : *Génération d'un Systme de transitions étiquetées à partir d'une spécification* B *événementiel*

Xavier Morselli (xavier.morselli@imag.fr) Marie-Laure Potet (marie-laure.potet@imag.fr)
Nicolas Stouls (nicolas.stouls@imag.fr)*
LSR-IMAG - 681, rue de la Passerelle, BP72, 38402 St Martin d'Hères Cedex

Le 13 avril 2010

Résumé

La source d'erreur la plus couteuse et la plus délicate à détecter dans un développement formel est l'erreur de spécification. Ainsi la premire phase d'un développement formel consiste généralement à représenter l'ensemble des comportements possibles sous la forme d'un automate. Partant de cette constatation, de nombreuses recherches portent sur la génération d'une machine B à partir d'une spécification telle qu'**UML**.

Cependant, il faut ensuite être capable de vérifier que la spécification respecte bien les comportements décrits. C'est pourquoi, nous avons réalisé un outil, GénéSyst, permettant d'extraire le contrôle d'un système B événementiel et de le représenter sous forme de système de transitions étiquetées. Le cas du raffinement est pris en compte et apparait sur l'automate produit sous la forme d'états hiérarchisés.

Cet outil est une implémentation des théories développées dans [SP04].

Mots clefs : Méthode B, spécification, raffinement, systèmes de transitions.

1 Fonctionnalits de GnSyst

L'approche B événementiel permet de modliser la fois les donnes, leur traitement et la dynamique d'un systme. Cette approche est base sur la notion d'vnements. Ceux-ci sont caractriss par une garde (une condition de dclenchabilit) et une action. La modlisation vnementielle introduit une difficult supplmentaire : le raisonnement sur la dynamique du systme. L'outil GénéSyst a pour but de permettre de visualiser cet aspect sous la forme de systmes de transitions symboliques. Il permet de :

- reprsenter par un tat un ensemble de valeurs (potentiellement infinis).
- calculer des transitions conditionnes reprsentant les hypothses sous lesquelles un vnement peut tre dclench (condition de dclenchabilit) dans un tat donn et les hypothses sous lesquelles un vnement permet d'atteindre un tat (condition d'atteignabilit).
- reprsenter le processus de raffinement par des systmes de transitions hirarchiss.

Le premier point permet de visualiser de manire lisible des systmes manipulant des donnes appartenant des domaines infinis ou grands.

---

0. Article publi dans les actes de la confrence AFADL'04 (http ://lifc.univ-fcomte.fr/afadl2004/)
*Ce travail est supporté par une bourse BDI confinancée par le CNRS et ST Microelectronics.

Le second point permet de prciser au mieux les conditions sous lesquelles une transition peut tre active. En particulier si elle n'est pas possible, toujours possible ou conditionne.

Enfin, la reprsentation des raffinements par des systmes de transitions hirarchiss permet de visualiser le lien entre deux niveaux de raffinement, en prcisant la structuration du systme de transitions abstrait. Le systme de transitions obtenu est complet et correct vis--vis de la spcification : il admet exactement les mmes traces que le systme B événementiel [SP04].

Ces travaux ont t initialement introduits par D. Bert et F. Cave [BC00]. Ils ont ensuite t tendus par S. Hamdane [Ham02], avant d'tre complts par N. Stouls et M-L Potet [SP04].

La figure 1 décrit un parking ayant un nombre de place limit ($NbPlaces$) et dans lequel les vhicules peuvent $entrer$ ou $sortir$. Un contrôleur, pour le moment inactif, ragit chacun de ces 2 stimulis externes. Sur les systèmes de transitions étiquetées produits, une condition est notée [ ] si elle est réductible à true et est note $[G]$ sinon. Ainsi, Les transitions $controler\_entree$ et $controler\_sortie$ ne sont pas conditionnes car cc est instanciée. En revanche, $entrer$ et $sortir$ dépendent de $NbVoit$ et sont donc conditionnées.

MACHINE $parking$
CONSTANTS $NbPlaces$
PROPERTIES $NbPlaces \in \mathsf{NAT} \wedge NbPlaces > 0$
VARIABLES $NbVoit, cc$
INVARIANT $(NbVoit \in 0..NbPlaces)$
$\wedge (cc \in -1..1)$
$\wedge (cc = -1 \Rightarrow NbVoit < NbPlaces)$
$\wedge (cc = 1 \Rightarrow NbVoit > 0)$
INITIALISATION $NbVoit := 0 \,||\, cc := 0$
OPERATIONS

entrer	$\widehat{=}$ SELECT $NbVoit < NbPlaces \wedge (cc = 0)$	THEN $NbVoit := NbVoit + 1 \,\|\, cc := 1$ END ;
controler_entree	$\widehat{=}$ SELECT $cc = 1$	THEN $cc := 0$ END ;
sortir	$\widehat{=}$ SELECT $NbVoit > 0 \wedge (cc = 0)$	THEN $NbVoit := NbVoit - 1 \,\|\, cc := -1$ END ;
controler_sortie	$\widehat{=}$ SELECT $cc = -1$	THEN $cc := 0$ END

END

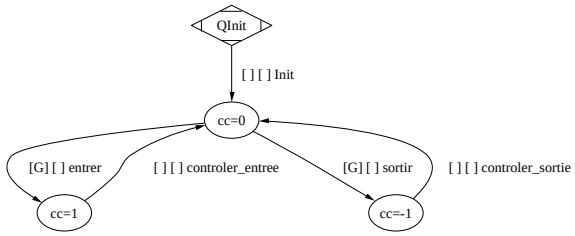

FIGURE 1 – Exemple d'un parking avec contrôleur (inactif) et son systme de transitions associ.

La figure 2, décrit un raffinement du parking de la figure 1, dans lequel un feu d'entrée a t introduit. La gestion de celui-ci est effectue par le contrôleur.

REFINEMENT $parking\_r1$
REFINES $parking$
SETS $Couleur\_feu = \{vert, rouge\}$
VARIABLES $feu, NbVoit, cc$
INVARIANT $feu \in Couleur\_feu$
$\wedge ((cc = 0 \wedge feu = vert) \Rightarrow NbVoit < NbPlaces)$
$\wedge ((cc = 0 \wedge feu = rouge) \Rightarrow NbVoit = NbPlaces)$
$\wedge (cc = 1 \Rightarrow feu = vert)$
$\wedge (cc = -1 \Rightarrow NbVoit < NbPlaces)$
$\wedge ((cc = -1 \wedge feu = rouge) \Rightarrow NbVoit = NbPlaces - 1)$
$\wedge ((cc = -1 \wedge feu = vert) \Rightarrow NbVoit < NbPlaces - 1)$
INITIALISATION $NbVoit := 0 || feu := vert || cc := 0$
OPERATIONS

entrer	$\widehat{=}$ SELECT $cc = 0 \wedge feu = vert$	THEN $NbVoit := NbVoit + 1 \,\|\, cc := 1$ END ;
controler_entree	$\widehat{=}$ SELECT $cc = 1$	THEN IF $NbVoit = NbPlaces$ THEN $feu := rouge$ END $\|\, cc := 0$ END ;
sortir	$\widehat{=}$ SELECT $NbVoit > 0 \wedge (cc = 0)$	THEN $NbVoit := NbVoit - 1 \,\|\, cc := -1$ END ;
controler_sortie	$\widehat{=}$ SELECT $cc = -1$	THEN IF $feu = rouge$ THEN $feu := vert$ END $\|\, cc := 0$ END

END

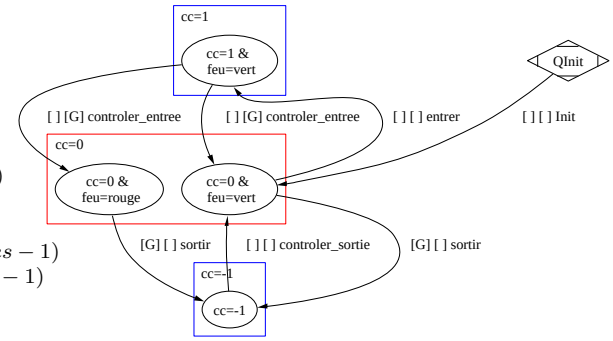

FIGURE 2 – Exemple du parking raffin et son système de transitions associé

2 Ralisation

GénéSyst prend en entre un systme B dcrit par une machine ou un raffinement. L'utilisateur fournit les tats du systme de transitions sous la forme d'une disjonction donnant les prdicats associs ces tats. Cette disjonction est fournie par l'intermdiaire de la clause ASSERTIONS. L'obligation de preuve associe cette clause va garantir que les tats reprsentent toutes les valeurs possibles de l'invariant.

Le rsultat est fourni sous la forme d'un fichier reprsentant le systme de transitions. Ce fichier, qualifi de *format intermdiaire*, est une reprsentation textuelle du systme de transitions produit. C'est partir de celui-ci que GénéSyst gnre des systmes de transitions dans diffrents formats exploitables par d'autres outils (pour l'instant, seuls les formats *DOT* et *BCG* sont supports).

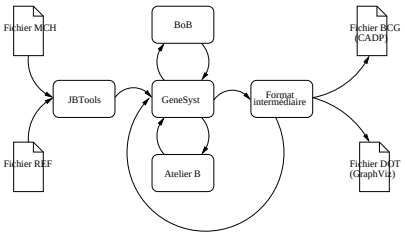

FIGURE 3 – Environnement de GénéSyst

GénéSyst fonctionne en mode automatique. Pour chaque vnement il cherche prouver si l'vnement est toujours dclenchable ou jamais dclenchable. Si une de ces preuves abouties, alors on a la rponse, sinon la transition est visualise par dfaut. Ceci ramne donc le problme de la recherche des conditions un problme de preuve. Le mme type d'approche est utilise pour l'atteignabilit. L'approche propose et sa correction est dcrite dans [SP04].

GénéSyst est ralis en Java en utilisant les outils suivants :

- le parseur JBTools [VTH02] qui permet d'analyser les spcifications.
- la **Bo**B (boite Outils B du LSR) qui permet de produire les obligations de preuve par calcul de plus faible pr-condition.
- le prouveur de l'*AtelierB* utilis actuellement en mode automatique.

3 Conclusions et perspective

GénéSyst peut tre utilis en phase de mise au point d'une spcification ou d'un dveloppement. Dans ce cadre, le systme de transitions peut tre modifi sans tre remis en cause totalement. GénéSyst peut donc aussi utiliser en entre une description (totale ou partielle) d'un systme de transitions. Dans ce cas son travail consiste vrifier et corriger ce systme de transitions. L'efficacit de l'outil est amliore puisque le systme de transitions permet de guider les preuves faire. De plus, l'utilisateur peut ainsi simplifier les conditions sur les transitions et GénéSyst produira les obligations de preuve assurant leur correction. Cette extension est en cours.

Une autre amlioration consiste dcoupler l'activit de preuve de l'outil. Ceci permet soit d'affiner les preuves de manire interactive avec le prouveur de l'*AtelierB* soit d'interfacer l'outil avec un autre dmonstrateur. La qualit du rsultat peut ainsi tre amliore puisque certaines transitions visualises correspondent un dfaut de preuve automatique.

D'un point de vue mthodologique, l'introduction d'tats hirarchiss permet de dcrire les tran-

sitions diffrents niveaux (entre les sur-tats ou entre les sous-tats). Ceci ncessite d'laborer des heuristiques permettant de faire ces choix de visualisation. Enfin, une interface graphique est en cours d'laboration. Le but est de pouvoir zoomer sur certaines parties du systme de transitions, par exemple pour voir le dtail d'une transition ou d'un tat.

GénéSyst est diffuse sur le site du LSR l'adresse :
http ://www-lsr.imag.fr/Les.Personnes/Nicolas.Stouls/

Références

[BC00] D. Bert and F. Cave. Construction of Finite Labelled Transition Systems from B Abstract Systems. In *Integrated Formal Methods*, volume 1945 of *LNCS*. Springer-Verlag, 2000.

[Ham02] Smaine Hamdane. Génération de systèmes de transition étiquetés à partir de la description d'un système d'évènements décrits avec le langage B. Rapport de maîtrise, Université Joseph Fourrier, Grenoble-1, France, mai 2002.

[SP04] N. Stouls and M.-L. Potet. Explicitation du contrle de dveloppements B vnementiel. In *AFADL'04*, LNCS. Springer-Verlag, 2004.

[VTH02] J.C. Voisinet, B. Tatibouet, and A. Hammad. jBTools : An experimental platform for the formal B method. In *PPPJ'02*, pages 137–140. Trinity College, Dublin, Ireland, Juin 2002.