

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

Xavier Morselli, Marie-Laure Potet, Nicolas Stouls

► **To cite this version:**

Xavier Morselli, Marie-Laure Potet, Nicolas Stouls. GénéSyst : Génération d'un système de transitions étiquetées à partir d'une spécification B événementiel. Jacques Julliand. *Approches Formelles dans l'Assistance au Développement de Logiciels (AFADL'04)*, Jun 2004, besançon, France. pp.317–320, 2004. <inria-00384212>

HAL Id: inria-00384212

<https://hal.inria.fr/inria-00384212>

Submitted on 13 Apr 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

GénéSyst : Génération d'un Système 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 coûteuse et la plus délicate à détecter dans un développement formel est l'erreur de spécification. Ainsi la première 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 apparaît 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 Fonctionnalités de GnSyst

L'approche B événementiel permet de modéliser la fois les données, leur traitement et la dynamique d'un système. Cette approche est basée sur la notion d'événements. Ceux-ci sont caractérisés par une garde (une condition de déclenchabilité) et une action. La modélisation événementielle introduit une difficulté supplémentaire : le raisonnement sur la dynamique du système. L'outil GénéSyst a pour but de permettre de visualiser cet aspect sous la forme de systèmes de transitions symboliques. Il permet de :

- représenter par un état un ensemble de valeurs (potentiellement infinis).
- calculer des transitions conditionnelles représentant les hypothèses sous lesquelles un événement peut être déclenché (condition de déclenchabilité) dans un état donné et les hypothèses sous lesquelles un événement permet d'atteindre un état (condition d'atteignabilité).
- représenter le processus de raffinement par des systèmes de transitions hiérarchisés.

Le premier point permet de visualiser de manière lisible des systèmes manipulant des données appartenant à des domaines infinis ou grands.

0. Article publié dans les actes de la conférence AFADL'04 (<http://lifc.univ-fcomte.fr/afadl2004/>)

*Ce travail est supporté par une bourse BDI cofinancée par le CNRS et ST Microelectronics.

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

Enfin, la représentation des raffinements par des systèmes de transitions hiérarchisés permet de visualiser le lien entre deux niveaux de raffinement, en précisant la structuration du système de transitions abstrait. Le système de transitions obtenu est complet et correct vis-à-vis de la spécification : il admet exactement les mêmes traces que le système B événementiel [SP04].

Ces travaux ont été initialement introduits par D. Bert et F. Cave [BC00]. Ils ont ensuite été tendus par S. Hamdane [Ham02], avant d'être complétés par N. Stouls et M-L Potet [SP04].

La figure 1 décrit un parking ayant un nombre de places limité ($NbPlaces$) et dans lequel les véhicules peuvent *entrer* ou *sortir*. Un contrôleur, pour le moment inactif, agit sur chacun de ces 2 stimuli externes. Sur les systèmes de transitions étiquetés produits, une condition est notée $[]$ si elle est réductible à true et est notée $[G]$ sinon. Ainsi, les transitions *contrôler_entrée* et *contrôler_sortie* ne sont pas conditionnées car cc est instanciée. En revanche, *entrer* et *sortir* dépendent de $NbVoit$ et sont donc conditionnés.

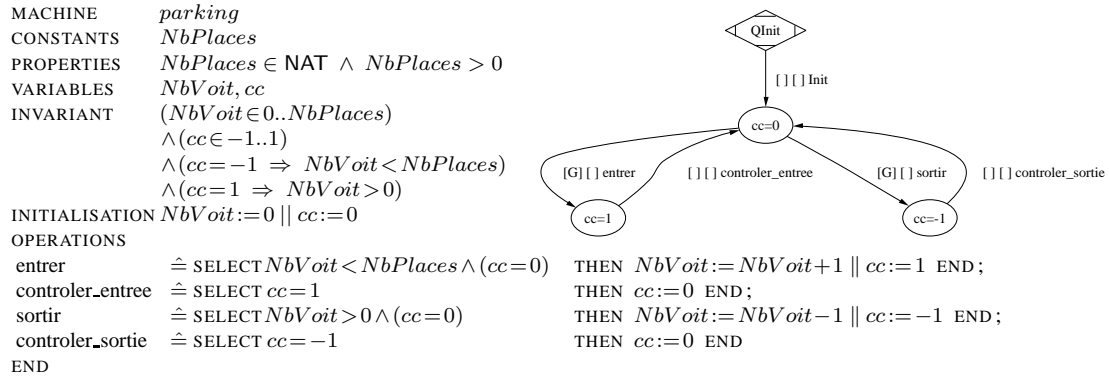


FIGURE 1 – Exemple d'un parking avec contrôleur (inactif) et son système de transitions associé.

La figure 2, décrit un raffinement du parking de la figure 1, dans lequel un feu d'entrée a été introduit. La gestion de celui-ci est effectuée par le contrôleur.

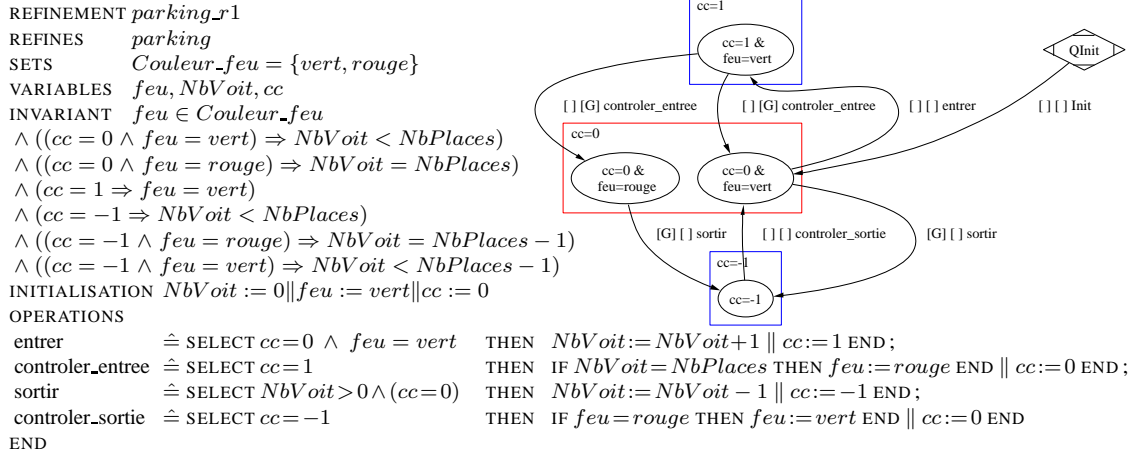


FIGURE 2 – Exemple du parking raffiné et son système de transitions associé

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 representent toutes les valeurs possibles de l'invariant.

Le rsultat est fourni sous la forme d'un fichier representant le systme de transitions. Ce fichier, qualifi de *format intermdiaire*, est une representation 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).

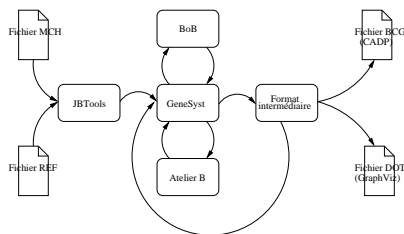


FIGURE 3 – Environnement de GénéSyst

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 sui-

vants :

- le parseur JBTools [VTH02] qui permet d'analyser les spcifications.
- la **BoB** (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 dcire les tran-

sitions différents niveaux (entre les sur-tats ou entre les sous-tats). Ceci nécessite d'élaborer des heuristiques permettant de faire ces choix de visualisation. Enfin, une interface graphique est en cours d'élaboration. Le but est de pouvoir zoomer sur certaines parties du système de transitions, par exemple pour voir le détail d'une transition ou d'un tat.

GénéSyst est diffusé 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 Fourier, Grenoble-1, France, mai 2002.
- [SP04] N. Stouls and M.-L. Potet. Explication du contrôle de développements 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.