



HAL
open science

Présentation des résultats du projet OpenEmbeDD

Charles André, Mariano Belaunde, Bernard Berthomieu, Christian Brunette, Agusti Canals, Hubert Garavel, Susanne Graf, Frederic Lang, Vincent Mahé, Michel Nakhlé, et al.

► **To cite this version:**

Charles André, Mariano Belaunde, Bernard Berthomieu, Christian Brunette, Agusti Canals, et al..
Présentation des résultats du projet OpenEmbeDD. Neptune, May 2009, Paris, France. inria-00381639

HAL Id: inria-00381639

<https://inria.hal.science/inria-00381639>

Submitted on 6 May 2009

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.

Présentation des résultats du projet OpenEmbeDD

Charles André (INRIA) Mariano Belaunde (FTRD) Bernard Berthomieu (LAAS)
Christian Brunette (INRIA) Agusti Canals (CS) Hubert Garavel (INRIA)
Susanne Graf (VERIMAG) Frédéric Lang (INRIA) Vincent Mahé (INRIA)
Michel Nakhlé (CS) Rémi Schneckenburger (CEA) Robert de Simone (INRIA)
Jean-Pierre Talpin (INRIA, coordinateur) François Vernadat (LAAS)

Résumé

OpenEmbeDD est une plate-forme générique, reposant sur les technologies d'Ingénierie Dirigée par les Modèles (IDM), et intégrant des outils d'aide à la conception d'applications temps réel embarqué (TR/E). Les technologies d'IDM utilisées dans OpenEmbeDD sont basées sur l'environnement Eclipse. Elles fournissent une solution simple et efficace pour l'intégration d'outils TR/E sur la plate-forme (par la réalisation d'un modeleur) et l'interopération de ses outils (par transformation de modèle). OpenEmbeDD peut être instanciée à un domaine d'application précis en sélectionnant, parmi l'ensemble des outils disponibles, ceux qui répondent à un cas d'utilisation particulier.

La plate-forme OpenEmbeDD s'installe en quelques clics. Elle fournit alors une solution d'IDM complète pour mettre en œuvre rapidement un modeleur et l'intégrer facilement à une suite d'outils. Pour cela, OpenEmbeDD repose sur le format EMF/Ecore afin de faciliter la manipulation de modèles, intègre le générateur d'éditeurs graphiques Topcased, le langage de transformation ATL, l'environnement de méta-modélisation Kermeta, l'éditeur de profils Papyrus.

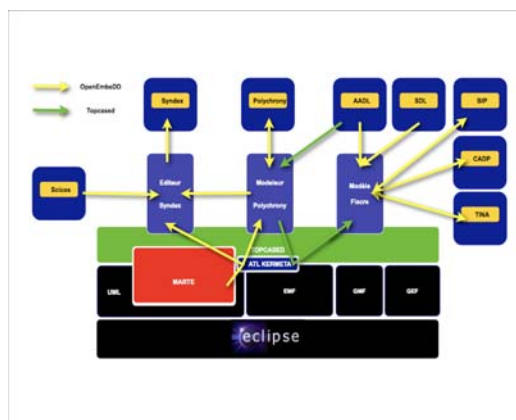
Sa conformité aux standards industriels les plus répandus est un avantage important. OpenEmbeDD intègre un modeleur UML complet accompagné de la mise en œuvre de référence du profil temps-réel MARTE. OpenEmbeDD inclus également d'autres standards comme AADL, SysML ou SDL.

La plate-forme OpenEmbeDD permet donc de compléter la modélisation d'applications utilisant des profils UML standards avec toute la souplesse et la spécificité d'outils métier d'aide à la conception pour le TR/E, et ce à moindre coût de développement.

Introduction

Le projet OpenEmbeDD compte 14 équipes de ses 8 partenaires industriels et académiques. Il a bénéficié d'un soutien important de l'ANR pour produire un effort de développement de 560 personnes*mois. Une clé de sa réussite de OpenEmbeDD a résidé dans la constitution d'une équipe chargée de l'intégration des fournitures de partenaires et leur livraison intégrée depuis le site web du projet <http://openembedd.org>. La qualité de cette intégration est manifeste dès l'installation, en quelques clics de souris, du paquetage téléchargé.

Plusieurs phases du projet se sont succédées et parfois chevauchées pour parvenir à ces résultats. La première a consisté en l'élaboration verticale de la plate-forme, par l'intégration des outils d'IDM puis par la réalisation des modeleurs Eclipse des outils d'aide à la conception considérés. Une seconde phase a consisté en la conception de passerelles pour permettre l'inter-opération de ces outils au moyen de transformations de modèles. L'ensemble de ses réalisations techniques a enfin été validé par une série d'études de cas qui ont permis d'en évaluer la pertinence technique.



Une plate-forme d'intégration

OpenEmbeDD est le premier projet plate-forme à bénéficier d'une cellule d'intégration et d'animation. Celle-ci a pour rôle de collecter les différents outils développés par les partenaires, logiciels et méta-modèles, et de les intégrer dans une suite logicielle homogène. La cellule assure le suivi des livrables et leur intégration dans la plate-forme. Elle réalise des tests « système » sur l'ensemble qui en résulte. Des outils ont été développés afin d'automatiser la construction et la gestion des tests.

La plate-forme OpenEmbeDD est basée sur l'Ingénierie Dirigée par les Modèles (IDM) et fournit notamment le langage ATL, l'environnement Kermeta, les modeleurs et les outils de générations de modeleurs Topcased. Elle a également vocation à accueillir tout nouvel outil (open source) en rapport avec l'IDM.

S'appuyant sur les modèles, un ensemble d'outils dédiés visent à faciliter la conception d'applications et de systèmes temps réel pour l'embarqué. Ils pourront être rejoints au fil du temps par d'autres outils de provenances variées, du monde académique comme de l'industrie.

Le caractère open source de la plate-forme lui confère un grand degré d'ouverture vis-à-vis de nouveaux outils. La suite logicielle étant bâtie à partir d'un *site-update* Eclipse complet, il est ainsi aisé de composer son propre environnement, au-dessus d'un Eclipse standard. Des solutions complètes *tout-en-un* sont également disponibles, pour Windows, Linux et MacOS X.

Outillage de l'ingénierie de modèles

La plate-forme du projet OpenEmBeDD a été conçue en couche afin d'offrir une méthodologie générique pour les besoins spécifiques du temps réel. L'approche suivie repose sur l'utilisation de l'Ingénierie Dirigée par les Modèles (IDM) dans l'environnement Eclipse. Les outils dédiés au temps réel étant alors soit conçus grâce à cette méthodologie, soit rendus compatibles avec celle-ci. En particulier, les efforts conjugués des partenaires ont permis d'intégrer un ensemble cohérent d'outils de modélisation génériques open source qui permettent de traiter la plupart des tâches d'une conception dirigée par les modèles. Elle s'articule autour de composants tels que le framework EMF de manipulation des modèles, le générateur de modeleurs graphiques Topcased, le langage de transformation ATL, l'environnement de méta-modélisation Kermeta, l'éditeur de profils Papyrus...

D'une part, la plate-forme offre des outils conformes aux standards industriels les plus répandus. Elle incorpore un modeleur UML complet, mais aussi les méta-modèles et profils AADL, SysML, MARTE, SDL. D'autre part, la plate-forme supporte aussi les approches basées sur l'utilisation de Langages Spécifiques au Domaine (ou DSL) et propose des composants vous permettant de construire et d'outiller votre propre langage. Toutes les activités de la création de DSL sont couvertes : de la conception de sa structure avec le modeleur Ecore, la spécification de sa sémantique opérationnelle avec Kermeta, à la génération automatique d'éditeur et de modeleur avec EMF et le générateur Topcased, en passant par les transformations de modèles avec ATL ou bien la vérification, ...

Offrant déjà une bonne couverture fonctionnelle pour faire de l'IDM, la plate forme est extensible et ouverte, et d'autres composants open source peuvent lui être ajoutés pour la compléter et l'adapter à vos besoins spécifiques.

Mise en œuvre du profil UML MARTE

OpenEmbeDD propose une large panoplie d'outils basée sur la modélisation normée par des standards industriels comme le langage UML.

Tout d'abord, la plateforme OpenEmbeDD fournit une implémentation complète de la spécification du profil MARTE. Le profil UML MARTE est une extension au modèle UML pour la modélisation et l'analyse des systèmes embarqués temps réel. Cette implémentation complète a évolué au fur et à mesure de la finalisation de la norme à l'OMG. Autour de ce profil, de nombreux outils ont été développés, comme l'outil TimeSquare. Cet outil permet d'éditer des contraintes temporelles dans un environnement de modélisation UML. Il permet également de vérifier des contraintes d'horloge, de générer des structures temporelles correctes et enfin d'afficher et explorer des chronogrammes au

format standard VCD. Le profil MARTE est également intégré entièrement aux modelers actuellement présent dans la plateforme OpenEmbeDD, Papyrus et TopCased.

Par ailleurs, des éditeurs pour différents diagrammes UML ont été générés grâce au framework TopCased. Cela comprend des éditeurs pour les diagrammes d'activité, les diagrammes de machine à état et enfin les diagrammes de séquence. De plus, un éditeur SDL a été ajouté à cette plateforme sous la forme d'un éditeur de diagramme de classes structurées profilé.

Enfin, la plateforme contient des outils permettant de modéliser le comportement d'une application en utilisant le langage UML2, sous la forme d'un éditeur de langage d'action paramétrable. A partir du comportement décrit en langage d'action et automatiquement traduit en diagrammes d'activité, l'utilisateur a la possibilité de générer du code C++ implémentant les algorithmes décrits. Des recherches ont également été menées au niveau de MARTE et la définition de composants sémantiques.

Méthodes formelles pour l'ingénierie

OpenEmbeDD fournit différents langages et outils spécifiques qui s'appuient sur des modèles de calcul parfaitement définis comme la modélisation synchrone et GALS, et la modélisation asynchrone.

Outils de modélisation synchrone

La plateforme OpenEmbeDD fournit une intégration complète de Polychrony¹ au travers de SME [2], son méta-modèle. Autour de SME, différents outils permettent de créer et manipuler des modèles conformes à SME graphiquement (diagrammes de flot de données, de contraintes d'horloges...) ou au travers d'un éditeur arborescent. Les services de vérification, de générations de code, et de simulation de Polychrony sont applicables à ces modèles au sein de la plateforme et ont été étendus afin de générer des modèles SME à partir des modèles existants de Polychrony. Enfin, des ponts permettent de passer des diagrammes d'état UML à SME et de SME vers Fiacre (ANR TopCased), le méta-modèle pivot pour l'asynchrone.

Gaspard [1], également intégré à la plateforme OpenEmbeDD, utilise le profil UML MARTE et plus particulièrement son sous-profil RSM, pour décrire les calculs parallèles de la partie logiciel, la structure parallèle de la partie matériel et les associations entre les deux. Ces modèles abstraits peuvent alors être déployés sur des implémentations spécifiques. Des transformations automatiques permettent de raffiner les modèles les plus abstraits vers les couches les plus basses selon les principes de l'IDM à des fins de simulations, de synthèses matériels, de validation formelle ou des calculs à haute performance.

La méthodologie AAA et le logiciel SynDEX² proposent des transformations de graphes afin d'optimiser la distribution et l'ordonnement d'applications sur des architectures distribuées, ce qui constitue la phase finale de la conception d'un système temps réel embarqué. Afin de tirer parti de cet outil, OpenEmbeDD intègre un méta-modèle de SynDEX, un modeler graphique permettant de manipuler des modèles conformes à celui-ci (diagramme d'application et d'architecture) et un générateur modèle vers texte. Différentes transformations permettent de passer à SynDEX par l'intermédiaire de Polychrony, de Gaspard2, ou d'un diagramme UML/MARTE.

Outils de vérification asynchrone

La plate-forme OpenEmbeDD permet d'exporter des modèles vers des outils de vérification formelle par le biais d'un langage intermédiaire développé dans le cadre du projet Topcased : FIACRE (*Format Intermédiaire pour les Architectures de Composants Répartis Embarqués*) [3,4]. Le langage FIACRE décrit des composants. Les composants élémentaires sont des machines à états étendues dont les transitions peuvent contenir des blocs de code structuré, permettant ainsi une modélisation compacte et une traduction aisée depuis les langages existants. Des composants plus complexes sont décrits hiérarchiquement par assemblage de composants plus simples dont les interactions peuvent être contraintes par des exigences temporelles et des priorités. Ainsi, des comportements complexes

1 <http://www.irisa.fr/espresso/polychrony>

2 <http://www-rocq.inria.fr/syndx>

peuvent être décrits de manière compositionnelle (la sémantique de l'ensemble est déduite de celles des constituants).

Plusieurs équipes ont développé des transformations depuis des modèles graphiques vers FIACRE. Ainsi, une transformation depuis AADL par l'équipe ACADIE de l'IRIT et une spécification de transformation depuis SDL a été réalisée par la société *Communication & Systems* en collaboration avec le groupe OLC du LAAS-CNRS, et une transformation depuis Signal/Polychrony a été développée par l'équipe ESPRESSO de l'INRIA Rennes Bretagne Atlantique. FIACRE est connecté à des outils de vérification formelle par le biais de deux outils de transformation intégrés à la plateforme OpenEmbeDD : FLAC et FRAC.

FLAC traduit les programmes FIACRE vers le langage LOTOS (une norme ISO pour la spécification de systèmes concurrents asynchrones), qui peut ensuite être vérifié avec CADP³ (*Construction and Analysis of Distributed Processes*) [5], une boîte à outils de vérification développée par l'équipe VASY de l'INRIA et utilisée par l'industrie du hardware pour vérifier des architectures et des processeurs de haute performance. FRAC traduit les programmes FIACRE vers le langage d'entrée de la boîte à outils TINA⁴ (*Time Petri Net Analyzer*) [6], développée par le groupe OLC du LAAS-CNRS. TINA opère sur une famille de modèles qui étendent les réseaux de Petri temporels et permet de vérifier par model-checking leur comportement discret et temporel.

Validation et cas d'étude

Des études de cas spécifiques à chacun des domaines d'application ciblés par OpenEmbeDD nous ont permis de valider la qualité des développements réalisés dans le cadre du projet.

Domaine télécom

L'expérimentation dans le domaine des télécommunications a consisté à construire un atelier pour le développement agile de services de télécommunication composites et intégrés, c'est-à-dire exploitant des briques issues aussi bien du monde des télécoms (par exemple l'envoi de SMS ou la localisation par GSM) que de l'internet (par exemple un service d'agenda en ligne). Pour ce faire on s'est appuyé sur des composants fournis par la plateforme OpenEmbeDD, tels que KerMeta, pour la simulation des modèles de services et SmartQVT pour la génération de code.

Cette expérimentation a impliqué d'exploiter en parallèle deux approches orthogonales mais complémentaires: l'approche SOA (Architecture orientée Services) et le MDE (Ingénierie des modèles). En pratique le SOA dans les Télécoms se traduit par la mise à disposition sous la forme de web services de composants de communication, appelés enablers. Le développement de nouveaux services se trouve grandement facilité pour des populations de programmeurs non expertes en réseau. La mise en œuvre du MDE se traduit par l'utilisation de formalismes de modélisation basés sur UML et par leur exploitation via des techniques de transformation de modèles et de génération de code.

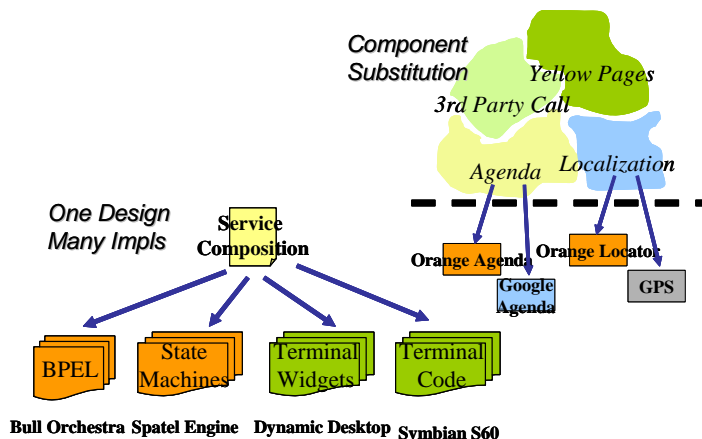
La pertinence de l'utilisation de l'ingénierie de modèles dans le développement de services telecom a été mis en évidence pendant les expérimentations. Le MDE agit en effet comme facilitateur pour:

- Traiter le problème de l'adaptation des interfaces de services aux multiples terminaux téléphoniques - au moins 5 grands types de systèmes d'exploitation sont aujourd'hui utilisés dans les smartphones du commerce. C'est en effet le fait de travailler à partir de modèles de services réutilisables dans différents contextes, que l'on va, via le développement de générateurs de code spécifiques, factoriser les efforts de développement.
- Gagner en fiabilité et en coût de maintenance, grâce à la possibilité de tester dès les premières phases du développement les logiques de service et de les faire évoluer. L'utilisation de modèles exécutables avec des rendus graphiques facilement lisibles et partageables sert également à la communication entre les différentes équipes en charge du développement.

3 <http://www.inrialpes.fr/vasy/cadp>

4 <http://www.laas.fr/tina>

La figure ci-contre illustre les deux niveaux de flexibilité que la combinaison SOA/MDE apporte au développement de services: d'une part une variabilité verticale (un design, plusieurs implantations générées), et d'autre part une variabilité horizontale, permettant le remplacement de composants ayant des fonctionnalités similaires, ceci via l'utilisation d'interfaces abstraites en UML (par exemple sans indications du protocoles de communication) et l'adaptation du code généré.

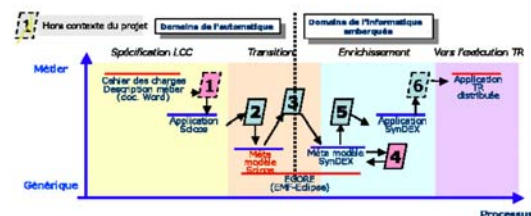


Domaine automobile

Le cas d'étude automobile traité dans OpenEmbeDD s'apparente aux techniques de planification et d'optimisation de mouvement des véhicules mises en oeuvre par PSA et expérimentées lors du projet RNTL2002 ECLIPSE [7]. Pour mémoire, le comportement dynamique des véhicules est un enjeu stratégique pour les constructeurs automobiles. A travers la notion de sécurité active, les constructeurs mettent l'accent non seulement sur la capacité du véhicule à suivre la trajectoire souhaitée par le conducteur, mais également sur ses capacités d'évitement d'obstacles. Le modèle complet utilisé pour le cas d'étude comporte 431 blocs Simulink, 24 états, et 12 ddl ; il est constitué de 4 blocs fonctionnels principaux :

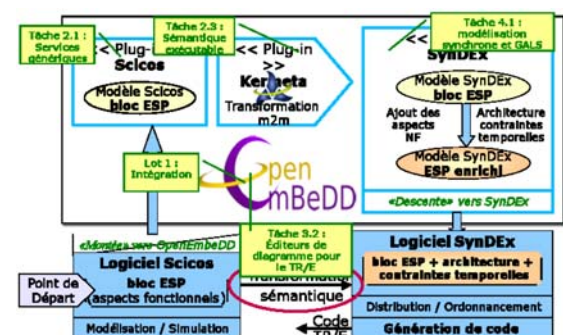
- Loi de Contrôle Commande (LCC) : DAE (direction assistée électrique), ABS (assistance au freinage), ESP (correction de trajectoire)
- Dynamique du véhicule : roues (vitesse de rotation des roues, couple de freinage et accélération), masses non-suspendues (pivot des roues), cinématique des trains + élasticité, châssis (caisse + suspension), pneu (modèle Pacejka)
- Stimulations (actions conducteur)
- Visualisation.

Les objectifs assignés à ce cas d'étude étaient de reproduire dans OpenEmbeDD le processus initial en utilisant les outils de la plate-forme OpenEmbeDD [8]. La procédure de validation consistait à comparer les deux approches sur un sous ensemble de la LCC PSA (ESP) et sa cible d'implantation (bi-proc/TCP). Le "design flow" associé à ce cas d'étude est représenté ci-contre.



Le déroulé du processus OpenEmbeDD est représenté par la seconde figure ci-contre. Les principales étapes en sont les suivantes :

- Spécification de la LCC sous Scicos à partir du cahier des charges
- Portage (via génération de XMI) vers le méta-modèle (MM) Scicos
- Transformation (via Kermeta) du MM-Scicos vers le MM-SynDEX
- Enrichissement du modèle par les aspects NF propres au MM-SynDEX (architecture et contraintes temporelles) via le plug-in Topcased-SynDEX
- Génération du fichier de format conforme à la grammaire de SynDEX



- Utilisation des fonctionnalités de distribution/ordonnancement et génération de code SynDEx.
- Les retours sur expérience constatés sont :
- Ergonomie du modelleur Topcased Ecore excellente ; fonctionnalité d'importation de Ecore vers Topcased très appréciable
 - Modèle SynDEx cible obtenu et complet ; bonne identification des concepts SynDEx ; extraction vers SynDEx très appréciable...
 - Équilibre de l'ergonomie entre manipulation par programmation et manipulation manuelle.
- L'outillage IDM de la plate-forme OpenEmbeDD s'est avéré riche, fiable et intégré ; le cas d'étude a été complètement validé et le processus associé à la conception de la LCC a été mené à bien.

Domaine avionique

Les expérimentations dans le domaine aéronautique ont été centrées sur l'évaluation de chaînes de développement continu orientées vers l'accès aux principaux outils de vérification.

Les systèmes embarqués sont des systèmes à logiciel prépondérant et sûrs de fonctionnement qui couvrent de nombreux domaines : communications, calcul numérique, gestion de bases de données, interface homme-machine. Mais du fait de leur criticité, ces systèmes sont fortement contraints en terme d'architecture: architecture modulaire IMA (Integrated Modular Avionics), ségrégation des partitions, et en terme de performance temps réel.

Les modèles vont comporter des représentations d'aspects fonctionnel, mais également physique et temporel, ainsi que comportemental. La notation unifiée MARTE permet de regrouper ces représentations dans des modèles de référence, permettant l'application d'outils de vérification, soit par une application directe, soit via la mise en œuvre de transformations Model To Model en ATL.

La notation unifiée permet de maîtriser le problème de la multiplicité des représentations nécessaires pour accéder aux différentes techniques et outils de vérification, en limitant les efforts de développement de ces représentations, ainsi que le coût du maintien de leur cohérence. L'automatisme des transformations permet de garantir la cohérence des modèles au cours des différentes étapes.

L'accès aux outils de vérification formelle TINA, SYNDEX, est important car leur application se situe dans les phases amont du cycle de développement et limite les hypothèses d'architectures potentielles.

Les outils de simulation permettent une analyse plus spécialisée, mais les transformations comme PERSIFORM prennent également du sens au vu de la complexité des formalismes à prendre en compte.

Les expérimentations ont montré que la part de MDE peut être favorisée dans les processus de développements spécifiques des systèmes critiques en respect de la norme DO178.

Références

- [1] A. Gamatié, S. Le Beux, E. Piel, A. Etien, R. Ben Atitallah, P. Marquet, and J.-L. Dekeyser. *A model driven design framework for high performance embedded systems*. Research Report 6614, INRIA, August 2008.
- [2] C. Brunette, J.-P. Talpin, A. Gamatie, and T. Gautier. *A metamodel for the design of polychronous systems*. Journal of Logic and Algebraic Programming, Elsevier, January 2009.
- [3] Bernard Berthomieu, Jean-Paul Bodeveix, Patrick Farail, Mamoun Filali, Hubert Garavel, Pierre Gauffillet, Frédéric Lang, and François Vernadat. *FIACRE: an Intermediate Language for Model Verification in the TOPCASED Environment*. In *Proceedings of the 4th European Congress on Embedded Real-Time Software ERTS'08* (Toulouse, France), 2008.
- [4] Bernard Berthomieu, Hubert Garavel, Frédéric Lang, and François Vernadat. *Verifying Dynamic Properties of Industrial Critical Systems Using TOPCASED/FIACRE*. In *ERCIM News* vol. 75, Octobre 2008.
- [5] Hubert Garavel, Frédéric Lang, Radu Mateescu, and Wendelin Serwe. *CADP 2006: A Toolbox for the Construction and Analysis of Distributed Processes*. In *Proceedings of the 19th International Conference on Computer Aided Verification CAV'2007* (Berlin, Germany). Lecture Notes in Computer Science vol. 4590, juillet 2007.

- [6] B. Berthomieu, P.O. Ribet, F. Vernadat, The tool TINA - Construction of Abstract State Spaces for Petri Nets and Time Petri Nets, *International Journal of Production Research*, Vol. 42, No 14, July 2004.
- [7] T. CAMBOIS : “ECLIPSE Project: Modeling and simulation of vehicle dynamics and control systems in Scicos”. 1st Scilab International Conference. December 2-3, 2004.
- [8] M. NAKHLE, C. FAURE : “De la modélisation/simulation à l’implantation temps réel de systèmes complexes”, “IliaTech : Modélisation, simulation et génération de code temps réel embarqué”. 17 décembre 2007. INRIA Paris-Rocquencourt, France.