



**HAL**  
open science

## Une plate-forme de test pour les cartes hybrides

Bertrand Gilles, Valérie-Anne Nicolas, Lionel Marcé

► **To cite this version:**

Bertrand Gilles, Valérie-Anne Nicolas, Lionel Marcé. Une plate-forme de test pour les cartes hybrides. plateforme AFIA 2003, Jul 2003, Laval, France. p.17. hal-00607386

**HAL Id: hal-00607386**

**<https://hal.univ-brest.fr/hal-00607386>**

Submitted on 8 Jul 2011

**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.

# Une plate-forme pour le test de cartes hybrides

Bertrand Gilles, Valérie-Anne Nicolas, Lionel Marcé

EA2215 - LIM1 / DÉPARTEMENT INFORMATIQUE Université de Bretagne Occidentale  
20 avenue Le Gorgeu - BP 809 - 29285 Brest Cédex, France  
{gilles|vnicolas|marce}@univ-brest.fr

**Résumé** : Nous présentons une méthode ainsi qu'une architecture et les premières briques d'une plate-forme pour le test de cartes hybrides. L'accent est mis sur la modélisation fonctionnelle des cartes et sur les stratégies de test fonctionnel développées pour le test en maintenance. Ces stratégies s'expriment en termes de contraintes et leur mise en oeuvre repose sur la programmation logique par contraintes. Pour finir, l'état actuel de la plate-forme et ses évolutions futures sont présentées.

**Mots-clés** : Test fonctionnel, génération automatique de données de test, cartes hybrides, test en maintenance, diagnostic.

## 1 Présentation de la méthode de test des cartes hybrides

Le test est une activité qui intervient tout au long du cycle de vie d'un composant. Le développement des technologies multi-média et des télécommunications, l'utilisation croissante de systèmes embarqués requièrent des composants matériels hybrides numériques/analogiques de plus en plus sûrs, et donc un processus de test adapté, fiable et robuste.

Dans ce contexte, il est crucial de mieux maîtriser les phases de test des cartes hybrides. Notre travail se situe dans ce cadre et concerne plus particulièrement le test en maintenance. Nous proposons une méthode de test de cartes hybrides : modélisation fonctionnelle de la carte, puis validation via des procédures de test fonctionnel et/ou l'utilisation d'outils de vérification. L'approche fonctionnelle, que ce soit au niveau modélisation ou au niveau du choix des stratégies de test est due à la spécificité du test en maintenance. Cette spécificité est de vérifier que certaines fonctionnalités de la carte ont le comportement attendu, sans pouvoir se baser sur une liste de fautes (structurelles) préétablie. Le test doit alors être basé sur des critères fonctionnels et non structurels. Un jeu de tests permettra ainsi de valider le comportement d'une carte par rapport à sa spécification fonctionnelle.

Une carte hybride est constituée de composants analogiques et de composants numériques. Les cartes hybrides que nous désirons tester ne font intervenir les composants analogiques que dans les interfaces d'entrée et de sortie de la carte. Les composants numériques constituent quant à eux le coeur de la carte, la partie contrôle. Ce type de cartes correspond à une part importante des cartes hybrides utilisées dans l'industrie. La méthode que nous présentons ici a pour but de tester spécifiquement ce type de cartes hybrides.

Malgré la coexistence de composants analogiques et numériques, le type de carte que nous voulons tester n'est pas un système hybride fortement couplé, car il n'y a pas de dynamique mixte discrète-continue sous-jacente. En effet, le système ainsi construit suit une dynamique continue dans ses interfaces et une dynamique discrète dans sa partie contrôle, sans qu'il y ait d'interférences profondes entre ces deux dynamiques. Nous avons par conséquent opté pour des modélisations relativement indépendantes des parties contrôle et interfaces. Notre démarche est la suivante : modélisation fonctionnelle des aspects numériques d'une part, modélisation fonctionnelle des aspects analogiques d'autre part, modélisation fonctionnelle globale de la carte en définissant un cadre commun dans lequel exprimer et faire interagir les deux modélisations précédentes.

La partie contrôle est modélisée directement sous forme d'automate, et plus précisément en tant que système séquentiel produisant des sorties (machine de Mealy). Les signaux analogiques sont abstraits par rapport au temps. Cette approche ne fait que remonter plus en amont la numérisation

effectuée par la carte. Les fonctions analogiques peuvent alors être également modélisées sous forme de machine de Mealy gérant ces signaux abstraits. Ainsi, la modélisation d'une carte hybride est composée d'un ensemble d'automates communiquants et un ensemble de signaux abstraits d'entrée.

Dans notre méthode, la validation d'une carte hybride repose principalement sur la génération automatique de données de test. Une *donnée de test* correspond à une représentation symbolique (non instanciée) de la donnée réelle. Dans notre cas, il s'agit d'un système de contraintes sur la donnée. Il est important de travailler sur une représentation symbolique des données pour la validité et l'efficacité de l'approche. L'instanciation des données de test n'est réalisée qu'au final, sur demande de l'utilisateur, pour fournir des données réelles à un banc de test par exemple.

Les stratégies de test fonctionnel utilisées peuvent être globales (concernant l'ensemble des comportements souhaités/effectifs de la carte) ou locales (à un comportement). La phase de test en maintenance intervient en effet le plus souvent à la suite d'une panne, d'un comportement erroné de la carte. Pour corriger la panne, il faut localiser le défaut à son origine. Il est donc nécessaire de disposer de stratégies de test local à un comportement. Dans notre modélisation, les différents comportements de la carte sont représentés par l'ensemble des chemins possibles dans un automate. Les différentes stratégies s'expriment donc en termes de parcours d'automates et de transitions, d'accessibilité d'états...

## 2 Plate-forme de test

La méthode résumée ci-dessus offre un cadre homogène pour la modélisation fonctionnelle, le test fonctionnel et la vérification de propriétés de cartes hybrides. D'un point de vue mise en oeuvre, cela se traduit par une plate-forme ouverte et homogène permettant la modélisation, le test et la vérification de cartes hybrides. Un prototype de cette plate-forme est en cours de réalisation. Actuellement, il permet de modéliser une carte en termes d'automates. Il permet également de concevoir et générer des données de test selon différentes stratégies de test fonctionnel local ou global.

À terme, la plate-forme comportera un module de discrétisation/reconstitution. Ce module permettra d'instancier les données de test par des valeurs réelles du domaine d'application de la carte, en se basant sur la modélisation de ses entrées. Dans un souci de complémentarité, la plate-forme offrira aussi des interfaces directes (via des modules de traduction automatique) vers des outils de vérification ou simulation comme Kronos ou Uppaal. De même, elle comportera un module de traduction automatique de notre formalisme de modélisation vers le langage VHDL-AMS afin d'assurer la compatibilité avec les interfaces des testeurs matériels (numériques, analogiques). Le langage VHDL-AMS est en effet un standard incontournable dans toute la chaîne de production de cartes. On le retrouve notamment comme format de modélisation dans les logiciels de CAO, ainsi qu'en format d'entrée de la plupart des simulateurs et bancs de test.

La partie concernant la mise en oeuvre des stratégies de test est réalisée dans un formalisme logique et en utilisant la programmation par contraintes, avec le solveur ECL<sup>i</sup>PS<sup>e</sup>. La programmation par contraintes est très pertinente pour la génération de données de test car elle permet de travailler sur des plages de valeurs de test dans un premier temps, que le testeur peut ensuite instancier à sa guise, en particulier pour prendre en compte des contraintes applicatives ou environnementales particulières. Par ailleurs, ce type de programmation très souple au niveau du mode des arguments pourrait permettre d'utiliser ces mêmes algorithmes de test dans un but de simulation. Ainsi, la plate-forme pourrait non seulement générer des données de test mais aussi simuler leur exécution sur la modélisation. Cela constituerait un complément appréciable lors de l'interprétation des résultats du test ainsi qu'une aide au diagnostic.

À plus long terme, il serait intéressant de généraliser notre approche afin de proposer des traitements spécifiques à différents types de cartes. Cela conduirait à des outils automatiques (ou semi-automatiques) de gestion des phases de test et maintenance, spécialisables par rapport à des types de composants/cartes précis (par reconnaissance de « pattern » ou apprentissage de comportement par exemple).

Ce projet bénéficie d'une coopération avec l'antenne brestoise de la société ISIS-MPP, spécialisée dans le test et l'outillage pour le test de cartes électroniques.