



# Génération de tests de conformité pour un système musical interactif temps-réel

Clément Poncelet Sanchez

► **To cite this version:**

Clément Poncelet Sanchez. Génération de tests de conformité pour un système musical interactif temps-réel. MSR 2013 - Modélisation des Systèmes Réactifs, 2013, Rennes, France. hal-00876652

**HAL Id: hal-00876652**

**<https://hal.inria.fr/hal-00876652>**

Submitted on 25 Oct 2013

**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.

---

# Génération de tests de conformité pour un système musical interactif temps-réel

## MSR'13 Modélisation des Systèmes Réactifs

**Clément Poncelet Sanchez**

*IRCAM*

*1 Place Igor-Stravinsky*

*75004 Paris*

*clement.poncelet@ircam.fr*

---

*RÉSUMÉ. Nous présentons une plateforme pour le test de conformité en approche boîte noire du système interactif temps réel Antescofo, qui est utilisé dans des concerts de musique mixte instrumentale-électronique. Les méthodes pour la génération et l'exécution des tests sont fondées sur des modèles du système sous forme de réseau d'automates temporisés, produits automatiquement à partir de partitions de musique mixte données et sur l'utilisation d'outil de la suite UPPAAL.*

*ABSTRACT. We present a framework for black box conformance testing of the realtime interactive music system Antescofo, which is used for mixed (instrumental-electronic) concerts. Our methods for test generation and executions are based on models of the system made of timed automata, which are computed automatically from given music scores. We use tools from the UPPAAL suite.*

*MOTS-CLÉS : Test de conformité, temps-réel, systèmes embarqués, automates temporisés, systèmes interactifs pour la musique mixte.*

*KEYWORDS: Conformance testing, realtime systems, embedded systems, timed automata, interactive music systems*

---

Le système interactif temps réel Antescofo<sup>1</sup> agit comme un « musicien électronique » lors de la production en concert de pièces de musique mixte électronique-instrumentale. Il fonctionne par couplage de deux modules. Un premier module dit *d'écoute artificielle* détecte en temps réel les événements instrumentaux de partition musicale et un tempo instantané, à partir du signal audio issu de l'instrument. Le second module est *réactif* : recevant les informations discrètes issues de la machine d'écoute, il envoie les actions de la partie électronique au moment spécifié (sous forme de messages à un environnement audio). La partition de musique mixte sur laquelle s'appuie le système à l'exécution est écrite dans un langage dédié de haut niveau comportant en particulier diverses primitives de synchronisation (entre les événements instrumentaux et électronique) et des stratégies de rattrapage d'erreurs<sup>2</sup>. Dans ce langage, les durées peuvent être exprimées en secondes ou en nombre de pulsation (*temps relatif*), se référant à la valeur de tempo inférée par le module d'écoute.

L'objectif de ce travail est la génération automatique de suites de test de conformité pour le module réactif d'Antescofo (dit implantation sous test, ou IUT), considéré comme une *boîte noire*. Les entrées et sorties *observables* de cette boîte noire sont respectivement les événements détectés par le module d'écoute et les actions électroniques du module réactif. De plus, le système étant *temps réel*, les délais sont aussi compris dans les données d'entrée et sortie. Nous avons fait le choix de suivre des approches de génération de test s'appuyant sur des modèles de réseaux d'automates temporisés, utilisant des extensions de l'outil de model checking UPPAAL<sup>3</sup>.

### Construction du modèle formel

Les tests sont générés dans le cadre d'une partition mixte donnée, à partir de laquelle un modèle formel du système est construit automatiquement. Nous considérons donc que cette partition contient suffisamment d'informations sur le comportement attendu du système pour s'affranchir d'intervention manuelle dans la construction du modèle. Nous avons contribué à la réalisation d'un compilateur produisant un réseau d'automates temporisés à partir d'une partition dans le langage dédié haut niveau d'Antescofo. En résumé, le réseau contient un automate modélisant les comportements possibles de l'environnement, c'est à dire le musicien (bien délimiter ce comportement est important pour éviter l'explosion du nombre de cas de test), et un automate par instrument électronique. Chaque automate du réseau a une horloge propre. Les horloges ont un écoulement synchrone, et peuvent être testées et remises à zéro par les transitions. Elles sont donc utilisées pour assurer les délais de la partition originale. Les communications entre automates sont faites par rendez-vous sur des symboles étiquetant les transitions et représentant les entrées (événements du musicien), les sorties (actions électroniques) et d'autres signaux internes non observables.

---

1. <http://repmus.ircam.fr/antescofo>

2. cf. Echeveste et al. *Formalisation des relations temporelles dans un contexte d'accompagnement musical automatique*, MSR 2011, et *Operational semantics of a domain specific language for real time musician-computer interaction*, JDEDS, 23 (4) 2013.

3. Hessel et al. *Testing Real-Time Systems Using UPPAAL*, FMT 2008.

Le compilateur est construit sur le parser du système Antescofo et parcourt l'arbre de syntaxe abstraite produit par ce dernier pour produire un code dans un langage intermédiaire bas niveau. Ce code est ensuite traduit au format d'entrée d'UPPAAL. De plus, des coordonnées graphiques y sont intégrées pour permettre une bonne visualisation des automates sur ce logiciel.

### Génération différée de suites de test

La méthode de test procède en deux étapes: la génération automatique d'une suite de tests à partir du modèle formel puis son exécution sur l'IUT avec verdict. L'extension d'UPPAAL COVER permet la génération de suites de tests exhaustives suivant des *critères de couverture* définis sous forme d'*observateurs*; intuitivement ce sont des machines à états finis décrivant le langage des transitions d'automates temporisés du modèle formel devant être visitées lors du test.

Une difficulté pour l'exécution des tests spécifique à notre cas d'étude est la présence de *multiples horloges* : une horloge (virtuelle) HM pour la synchronisation des automates temporisés composant le modèle formel, l'horloge HD définie par le tempo détecté par la machine d'écoute pour l'IUT et enfin l'horloge physique (HP), en secondes, accessible sur l'OS de la machine exécutant les tests.

Pour l'exécution d'une suite de tests  $T$  produite par COVER, nous avons considéré que  $HM = HD$ . Les entrées de  $T$  sont ajoutées avec leurs délais à la partition Antescofo originale, pour être "jouées" par l'IUT dans un thread séparé. Tous les délais d'entrées et sorties sont alors exprimés en temps relatif (HD), ce qui permet de comparer les délais de sortie de  $T$  avec ceux de l'IUT. Grâce à la couverture formelle garantie par COVER, cette approche est adaptée au "génie logiciel", pour déboguer.

Nous avons développé une seconde approche plus destinée à la préparation de concerts, où HM est définie par une *courbe de tempo* simulant le comportement d'un musicien. Les suites de tests sont produites à partir de telles courbes et du modèle.

### Perspectives

Nous envisageons de développer une méthode ad'hoc de génération adaptative à la volée de valeurs de test, les outils d'UPPAAL pour le test à la volée n'étant pas adaptés à notre cas, en raison des multiples horloges. Une autre perspective est la génération différée de partitions de test «extrêmes», voir d'entrées audio, pour tester d'autres aspects du système.