
Stimulus : un langage de programmation synchrone à contrainte appliqué à la simulation d'exigences temps-réel fonctionnelles

Bertrand Jeannet*†¹

¹3DS – Dassault Systèmes – France

Résumé

Les langages synchrones tels que Lustre et Lucid Synchrone ont été conçus pour programmer des systèmes temps-réels de contrôle-commande de manière sûre et efficace. Pour exprimer des propriétés sur ces programmes a ensuite été introduite la notion d'observateur synchrone. Enfin, pour simuler et tester ces programmes, le besoin a été identifié de modéliser des environnements non-déterministes pour stimuler le programme sous test avec des entrées réalistes, ce qui a mené notamment au langage à contraintes Lutin et à la machine d'exécution Lurette.

Stimulus synthétise ces travaux : il combine les contraintes de Lutin et les automates hiérarchiques de Lucid Synchrone, et il introduit une notion d'observateur permettant de transformer tout fragment de programme générant des signaux en reconnaisseur de propriété sur ces signaux. Dans ce cadre, exécuter un programme consiste, à chaque pas d'exécution synchrone, à faire évoluer l'état des automates et à résoudre les contraintes actives sur les signaux à l'aide d'un solveur.

L'application privilégiée qui a guidé nos travaux est la simulation des exigences fonctionnelles d'un système temps-réel, que nous illustrerons.

*Intervenant

†Auteur correspondant: Bertrand.JEANNET@3ds.com