Un langage de contexte de preuve pour la validation formelle de modèles logiciels

Philippe Dhaussy[*] — Julien Auvray*
Stéphane De Belloy** — Frédéric Boniol*** — Eric Landel[* +]

* Laboratoire DTN, ENSIETA, BREST, F-29806 cedex 9
{dhaussy, auvrayju, landeler} @ensieta.fr
http://www.ensieta.fr/dtn
** THALES AIR SYSTEMS, BP 20351 94628 RUNGIS Cedex
stephane.debelloy@fr.thalesgroup.com
*** IRIT-ENSEEIHT, 2 rue C. Camichel BP 7122 – F-31071 Toulouse, cedex 7
frederic.boniol@enseeiht.fr
+ CS-SI, 6, avenue Saint Granier, Toulouse

Résumé. Pour améliorer les pratiques dans le domaine de la validation formelle de modèles, nous explorons un axe de recherche dans lequel nous formalisons la notion de « contexte de preuve » intégrant la description du comportement de l'environnement interagissant avec le modèle et les propriétés à vérifier dans ce contexte. L'article présente le langage CDL (*Context Description Language*) proposé à l'utilisateur pour la description des contextes de preuve. Ceux-ci sont exploités, actuellement dans nos travaux, par une technique de vérification de type *model-checking* avec la mise en œuvre d'observateurs. Dans une approche Ingénierie Dirigée par les Modèles (IDM), les modèles de contextes sont transformés en modèles d'automates temporisés puis en codes exploitables par l'outil OBP/IFx (*Observer-Based Prover*). Ce travail a donné lieu à plusieurs expérimentations industrielles comme la validation formelle d'un protocole de communication avionique pour l'AIRBUS A380. Dans cet article, nous décrivons l'application de notre approche pour la validation d'un modèle de contrôleur de système aérien conçu par THALES. L'article rend compte de la mise en œuvre du langage CDL et d'un retour d'expérience.

1 Introduction

La validation formelle des architectures logicielles. Dans le domaine des systèmes embarqués, les architectures logicielles doivent être conçues pour assurer au sein de ceux-ci des fonctions de plus en plus vitales. Les architectures de calculateurs comme ceux des domaines avioniques ou automobiles, les systèmes d'informations critiques ou d'acquisition de données sont soumis à des contraintes de temps et de fiabilité très importantes. Leur développement nécessite donc des techniques d'ingénierie prenant en compte ces caractéristiques dès les phases amont de leur cycle de vie. Par exemple, en ce qui concerne la gestion des exigences, spécifiées suite à l'expression du besoin des utilisateurs, et exploitées