

Les Zero-Safe Nets pour la Préservation de la TTC dans les Diagrammes d'Activité d'UML

Sabine Boufenara*, Faiza Belala**, Chafia Bouanaka***

*Université Benhamouda de Jijel, Ouled Aissa, Algérie
sabineboufenara@yahoo.com

Université Mentouri de Constantine, Route de Ain Elbey, Algérie
{**belalafaiza@hotmail.com
***c_bouana@yahoo.fr}

Résumé. Avec les extensions d'UML 2.0, les réseaux de Petri, utilisés comme cadre sémantique formel pour les diagrammes d'activité d'UML, ne permettent plus d'exprimer les nouvelles constructions de haut niveau telles que la traverse-to-completion. Cette dernière nécessite une synchronisation globale entre les nœuds fork et join, totalement absente dans les réseaux de Petri ordinaires offrant une synchronisation locale. Afin de préserver le comportement des diagrammes d'activité, nous proposons l'adoption des réseaux de Petri zero-safe, une classe particulière des réseaux de Petri. Nous définissons un passage générique des diagrammes d'activité d'UML vers cette classe des réseaux de Petri. La formalisation que nous proposons assure la préservation de la sémantique opérationnelle des diagrammes d'activité en mettant l'accent sur le principe de la traverse-to-completion et la synchronisation des nœuds fork et join supportant ainsi les flots de contrôle et donnée et la concurrence des threads déclenchés par le noeud fork.

1 Introduction

Le langage UML (*Unified Modelling Language*) (Clark et Evans, 2000) est devenu un standard industriel pour la modélisation des systèmes. C'est un langage visuel à caractère expressif décrivant la structure et le comportement des systèmes orientés objet. Il a une sémantique semi-formelle car celle-ci est spécifiée sous la forme d'un méta-modèle (diagramme de classe) et de règles OCL (OCL : Object Constraint Language) pour la syntaxe abstraite et la sémantique statique, et de commentaires en langage naturel pour la sémantique dynamique. Et de ce fait, il manque de capacités d'analyses et d'automatisation ce qui le rend mal adapté à la simulation, etc. Adopter des modèles formels n'est pas la solution idéale car ils seront mal acceptés par des utilisateurs novices d'UML. La solution adoptée par plusieurs chercheurs est de définir une vraie sémantique d'UML en conservant sa syntaxe, car proche de l'utilisateur et définir des règles de transformation des diagrammes d'UML en des modèles formels pour une éventuelle analyse, simulation ou autres.

Parmi les travaux de formalisation d'UML, nous pouvons mentionner celui de Kim et Carrington (1999) qui propose une formalisation du diagramme de classes basée sur le lan-