

Résumé

UML, un standard de fait pour la modélisation orientée objet, subit des reproches incessants sur le manque de ses fondements formels. Cette notation, très riche en éléments de modélisation, souffre encore d'une carence sémantique sur bien nombre de ses concepts. Son imprécision complique la tâche de spécification et rend les techniques de preuve et de vérification, extrêmement difficiles, voire impossibles, quand la correction de la modélisation devient une condition incontournable en génie logiciel. Reconnues pour leur fondements théoriques rigoureux, les techniques de spécification formelle sont à même de répondre à ce besoin ; elle sont toutefois peu utilisées en raison de leur complexité. Aussi, pour rallier convivialité et rigueur, nous proposons une méthodologie de spécification et de vérification des diagrammes dynamiques UML selon une approche orientée valeur, basée sur les réseaux de Petri à Objets et la logique temporelle. Nous nous intéressons plus particulièrement aux systèmes communicants et retenons à cet effet, les diagrammes d'activités partitionnés et les diagrammes d'états-transitions connectés au moyen des diagrammes de collaboration. Nous commençons par définir la technique de transformation des diagrammes dynamiques UML en réseaux de Petri à Objets. Les diagrammes d'objets et de séquence sont ensuite exploités pour produire les objets et les messages nécessaires à l'initialisation des réseaux de Petri. Pour vérifier que les modèles construits préservent les propriétés du système, nous recourons au langage OCL, une partie d'UML pour l'expression de contraintes sur les modèles. Les contraintes OCL sont traduites en formules de logique temporelle pour être contrôlées sur les réseaux de Petri Objets dérivés des diagrammes UML. Pour assurer cette validation, la modélisation de flot d'objets sur les diagrammes d'états-transitions et d'activités partitionnés est requise. Elle consiste en la création ou destruction d'objets des extrémités d'association. Cet intérêt pour le mouvement des objets à travers les extrémités d'association provient de la structure des expressions OCL qui repose essentiellement sur les collections d'objets. Enfin, notre préoccupation majeure étant de proposer une méthodologie de spécification ralliant entre convivialité et fiabilité en épargnant au modélisateur la contrainte de verser dans les langages formels, une fois les résultats du model checking rendus, nous proposons de les analyser et de les interpréter pour les retourner au modélisateur UML dans une forme qui lui soit lisible.

Abstract

UML, the de-facto standard for object-oriented modeling, currently still lacks a rigorously defined semantics for its models. This makes formal analysis and verification of model properties, extremely difficult. To verify if UML models meet the system required properties, we resort to the Object Constraint Language OCL which is part of UML for the expression of constraints over models. Thus, the main purpose of this thesis is to automate the formal validation of UML state-based models, according a value-oriented approach. The marriage of Object Petri nets with temporal logics seems a suitable formalism for translating and then validating these models. We are especially interested in the modeling of communicating systems and so we focus on the behavior of systems expressed in UML using activity partitions or statecharts connected by means of collaboration diagrams. We first, consider how these diagrams can be transformed into Object Petri Nets to formalize the object dynamics, in an object-oriented context. Secondly, we resort to the object and sequence diagrams which provide respectively identified objects and events for initializing the Object Petri nets derived from the statecharts and activity partitions. Thirdly, we translate the OCL invariants into temporal logic properties so that they can be checked on the Petri nets derived from the UML models. To deal with this checking, integration of object flows within the statecharts and activity partitions is required. These object flows express the dynamic creation and deletion of objects in the class association ends. Our interest in the association ends is motivated by the fact that they constitute the most important constructs of OCL expressions. Finally, since the methodology supposes that the designer does not master the formal specification, the analysis results rendered by the model checker, will not be obviously meaningful to him. So, we propose to automate the analysis of these results as well as their interpretation on the UML models so that they could be presented to him in a form where the error in models is simply and clearly pointed out.
