

Titre: *Formal Object Oriented Development of Software Systems
using LOTOS*

Résumé

Les méthodes formelles sont nécessaires pour construire des logiciels corrects: c'est-à-dire, des logiciels qui répondent aux spécifications. Les spécifications formelles sont non ambiguës et exploitables. La construction d'un modèle formel améliore la compréhension. La modélisation du non-déterminisme et son élimination nécessitent des étapes formelles et elle permet de rendre les décisions de conception et d'exécution moins ad hoc. Les modèles formels offrent la possibilité d'effectuer des transformations mathématiques, de raisonner et ils facilitent les procédures rigoureuses de test. Cependant, les méthodes formelles ne sont pas bien utilisées pour le développement des logiciels. Dans la plupart des cas, la raison principale est le manque d'outils de développement. En plus, beaucoup d'ingénieurs ne reconnaissent pas le besoin de rigueur.

Les techniques orientées objets sont un grand succès dans la production de système logiciels complexes. Ces méthodes sont fondées sur des modèles mathématiques d'abstraction et de classification. L'approche orientée objet offre une cohérence conceptuelle pour trouver des cycles de développement de logiciel. L'inhérente flexibilité des approches objets peut conduire à un style incrémental et interactif pour le développement, une conséquence pouvant être l'insuffisance de rigueur. Ce problème est renforcé par un manque de sémantiques des concepts orientés objets.

Les méthodes formelles et orientées objets sont complémentaires dans le cycle de développement du logiciel : les méthodes orientées objets peuvent être utiles pour gérer la construction des modèles formels et le formalisme peut ajouter de la rigueur au développement de logiciel fondé sur l'approche orientée objet. Cette thèse montre comment le développement formel orienté objet peut opérer sur l'analyse et la capture de besoins à la conception et à l'implantation.

Un langage formel d'analyse orientée objet est défini sur la sémantique d'un système de transition d'état. Ce langage est dit orienté client : quelques vues graphiques des relations orientées objets dans les modèles formels d'analyse sont présentées et les spécifications produites précisent ce qui est demandé plutôt que comment les besoins sont satisfaits. Une traduction vers ACT ONE fournit un modèle exécutable pour la validation d'un client. Cette validation est fondée sur un énoncé précis de la relation entre classes et types (et sous classes et sous types). La structure du modèle résultant en ACT ONE correspond à la structure du domaine du problème comme transmis au client.

Le passage de l'analyse à la conception demande une extension du modèle de spécification afin d'inclure une sémantique pour la communication des objets. Une algèbre de processus fournit un modèle formel pour la spécification des propriétés de communication.

LOTOS, qui combine ACT ONE et une algèbre de processus dans un modèle sémantique cohérent fournit un moyen de construire une sémantique de conception orientée objet. La conception est définie comme le processus de transformation d'un modèle de spécification orientée client en une conception orientée implantation, tout en maintenant la correction. Les transformations préservant la correction sont définies pour transférer la structure de spécification en structure de conception, pour manipuler la structure de conception et changer les modèles de communication interne.

La conception doit être réalisée en fonction d'un environnement particulier d'implantation. Cette thèse fait en résumé plusieurs environnements différents pour l'implantation des conceptions orientés objets LOTOS. Cela illustre l'importance de la compréhension de la sémantique du langage de programmation. Nous montrons comment Eiffel peut être utilisé pour mettre en oeuvre nos idées.

Une étude de cas évalue les modèles formels orientés objets et les méthodes. Cela met en évidence toutes les étapes de développement du logiciel et souligne le rôle de la structuration : elle améliore la compréhension et la communication et rend la validation et la vérification plus facile et meilleure.

Cette thèse montre que la technologie formelle orientée objet est prête pour un transfert vers l'industrie. Les méthodes doivent être exploitées le plus tôt possible : le développement orientée objet peut inclure les méthodes formelles sans coût significatif et les méthodes formelles peuvent utiliser le paradigme objet pour gérer la complexité. Cette thèse donne une métaméthodologie pour le développement formel orientée objet et un ensemble d'outils conceptuels qui fait du développement de systèmes logiciels une discipline entière pour les ingénieurs.