

Étude de la changeabilité des systèmes orientés objet

Stéphane Vaucher*, Houari Sahraoui*

*Département d'informatique et de recherche opérationnelle,
Université de Montréal
CP 6128 succ. Centre-Ville, Montreal, Quebec, H3C 3J7, Canada
vauchers,sahraouh@iro.umontreal.ca

Résumé. Plusieurs études montrent qu'avec le temps, la plupart des systèmes deviennent difficiles à maintenir et que leur croissance ralentit. Il existe cependant certains systèmes qui utilisent les mécanismes fournis par le paradigme des objets pour soutenir un rythme de développement élevé. Dans cet article, nous étudions les facteurs qui affectent la changeabilité de quatre logiciels libres populaires. Deux applications et deux librairies ont été sélectionnées, puis caractérisées avec des métriques orientées objet classiques. Ces informations ont été utilisées pour bâtir des modèles de prédiction de changement avec des techniques d'apprentissage automatique. Dans le cas de deux librairies avec des modèles de domaine suffisamment précis, les modèles prédictifs ont été capables d'estimer correctement le taux de changement dans le code. Dans le cas de deux applications, ces modèles étaient beaucoup moins précis, mais il a été toutefois possible de prédire les changements dans les classes responsables des interfaces graphiques.

1 Introduction

Nous nous intéressons à l'étude des facteurs qui peuvent influencer l'amplitude des changements dans le code lors de l'évolution des logiciels à objets. En effet, la maintenance des logiciels représente une dépense majeure pour l'industrie (Bell, 2000; Hamlet et Maybee, 2000). Cependant, malgré des avancées importantes en génie logiciel lors des dernières décennies, il est encore difficile d'estimer les coûts de maintenance. Cette situation est d'autant plus préoccupante qu'il existe beaucoup plus de systèmes en maintenance que de nouveaux systèmes développés.

Une façon concrète d'estimer les coûts de maintenance consiste à déterminer a priori, à chaque étape de l'évolution du logiciel (i.e. à chaque version), la quantité de code à changer pour répondre à un ensemble de besoins (Khoshgoftaar et Szabo, 1994; Khoshgoftaar et al., 1996; Nagappan et Ball, 2005). La prédiction de changements dans le code n'est toutefois pas une tâche aisée. En effet, l'état actuel des connaissances ne permet pas de connaître les facteurs qui peuvent influencer positivement ou négativement l'amplitude des changements. Une approche empirique est une bonne alternative, car elle permet d'étudier ces influences en se basant sur des données historiques. Jusqu'à récemment, elle a été utilisée à petite échelle sur des données provenant de systèmes uniques industriels (Li et Henry, 1993; Nagappan et