

Evaluation de l'efficacité des implémentations de l'héritage multiple en typage statique

Floréal Morandat*, Roland Ducournau*, Jean Privat**

*LIRMM – CNRS et Université Montpellier II
161 rue Ada Montpellier – 34392 Cedex 5 France
{morandat,ducour}@lirmm.fr,
<http://www.lirmm.fr/~{morandat,ducour}/>

**Université du Québec à Montréal
privat.jean@uqam.ca
<http://www.info2.uqam.ca/~privat/>

Résumé. La programmation par objets présente une apparente incompatibilité entre trois termes : l'héritage multiple, l'efficacité et l'hypothèse du monde ouvert — en particulier, le chargement dynamique. Cet article présente des résultats d'expérimentations exhaustives comparant l'efficacité de différentes *techniques d'implémentation* (coloration, BTM, hachage parfait, ...) dans le contexte de différents *schémas de compilation* (de la compilation séparée avec chargement dynamique à la compilation purement globale). Les tests sont effectués avec et sur le compilateur du langage PRM. Ils confirment pour l'essentiel les résultats théoriques antérieurs tout en montrant une sur-additivité marquée des surcoûts. Les schémas d'optimisation globale démontrent un gain significatif par rapport à la coloration qui fait fonction de référence. Des techniques comme la simulation des accesseurs ou le hachage parfait entraînent un surcoût limité, mais la combinaison des deux double le surcoût total.

1 Introduction

L'hypothèse du monde ouvert (OWA pour *Open World Assumption*) représente certainement le contexte le plus favorable pour obtenir la *réutilisabilité* prônée par le génie logiciel. Une classe doit pouvoir être conçue, programmée, compilée et implémentée indépendamment de ses usages futurs et en particulier de ses sous-classes. C'est ce qui permet d'assurer la compilation séparée et le chargement dynamique.

Cependant l'héritage multiple — ou ses variantes comme le sous-typage multiple d'interfaces à la JAVA — s'est révélé difficile à implémenter sous l'hypothèse du monde ouvert. En typage statique, seule l'hypothèse du monde clos (CWA) permet d'obtenir la même efficacité qu'en héritage simple, c'est-à-dire avec une implémentation en temps constant nécessitant un espace linéaire dans la taille de la relation de spécialisation. Les deux langages les plus utilisés actuellement illustrent bien ce point. C++, avec