6th International Conference on Applied Informatics
Eger, Hungary, January 27–31, 2004.

A generative approach for the family polymorphism problem in C++

István Zólyomi, Zoltán Porkoláb

Department of Programming Languages and Compilers,
Eötvös Loránd University, Faculty of Informatics
e-mail: {scamel, gsd}@elte.hu

Abstract

Family polymorphism – strongly investigated by Erik Ernst and others – takes traditional polymorphism to the multi-object level. The object-oriented paradigm provides safe and flexible use of objects of classes arranged to inheritance hierarchies. Late binding ensures that we use the appropriate function body when we call a method on an actual object via polymorphic reference. In the same time we have compile-time guarantees to use only valid calls.

The problem arises when we use two or more independent hierarchies of classes together. In this case the collaborating "families" may consist of similar but not interchangeable classes. Because there can be subtype relationship between classes in the different groups, it is not obvious to implement a constraint ensuring that only classes of the same family are used together. Traditional object-oriented languages are not able to handle this situation. Proposed solutions vary from run-time assertions to extensions of existing programming languages (like gbeta). In this paper we present a generative solution to express such constraints in the C++ language using templates. We use only standard C++ language features in our implementation.

Categories and Subject Descriptors:
D.1.5 [Programming Techniques]: Object-oriented Programming;
D.1.m [Programming Techniques]: Miscellaneous

Key Words and Phrases: Family polymorphism, Generative programming