7^{th} *International Conference on Applied Informatics*
Eger, Hungary, 2007.

Supporting generic programing in CORBA IDL

Ilir Kurti[a], **Zoltán Porkoláb**[b]

[a]Universiteti Aleksandër Moisiu, Durrës
e-mail: ikurti@uamd.edu.al

[b]Department of Programming Languages and Compilers,
Faculty of Informatics, Eötvös Loránd University, Budapest
e-mail: gsd@elte.hu

Abstract

The Common Object Request Broker Architecture (CORBA) is a standard that enables software components written in multiple computer languages and running on multiple platforms to interoperate. CORBA uses the interface definition language (IDL) to specify the interfaces between components. Portability is achieved through the concept of language mapping: the IDL compiler generates stub and skeleton code that the client and servant links to. Generic programming is an emerging paradigm that focuses on abstracting types to a narrow collection of functional requirements and on implementing algorithms in terms of these requirements. Generic programming is implemented in ADA via generics, and is essential part of the C++ programming language through templates. Generics are also introduced recently in other languages, like Java and C#. It is an unfortunate case that CORBA IDL does not support generics, which causes unnecessary code repetition both in declaration level in IDL and implementation level in stubs and skeletons. In this article we discuss the advantages of introducing templates into the IDL language. IDL<T> – a prototype IDL compiler supporting templates – is presented. With IDL<T> interface definition is more obvious, and several code repetition cases could be eliminated at implementation.

Keywords: CORBA, IDL, Generic-programming

MSC: 68M14 Distributed systems

Ilir Kurti
Universiteti Aleksandër Moisiu, Lagja 1, Rr. Currilave, Durrës, Albania

Zoltán Porkoláb
Eötvös Loránd University, Faculty of Informatics, Dept. of Programming Languages and Compilers, Pázmány Péter sétány 1/c., H-1117 Budapest, Hungary