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

Clean-CORBA Interface Supporting Skeletons *

Zoltán Hernyák[1], Zoltán Horváth[2], Viktória Zsók[2]

[1] Department of Information Technology
Eszterházy Károly College
e-mail: aroan@ektf.hu

[2] Department of Programming Languages and Compilers
Eötvös Loránd University, Budapest
e-mail: hz@inf.elte.hu, zsv@inf.elte.hu

Abstract

The present Clean-CORBA interface supports only in a limited way developing and using skeletons for distributed and cluster computing. The interface user encounters three major problems. Clean skeletons [1] are parameterized by types while the IDL compiler does not support generating stubs for polymorphic functions. The other major problem is that there is a need for such a component of the middleware system which starts the skeleton according a strategy given as parameter. The third problem is that the actual Clean-CORBA is not able to use threads and asynchronous communication [2].

The IDL type any is not a complete solution for the first problem. It allows data types to be examined and extracted using the type code interface, but it causes run-time overhead and needs language support in Clean to use a newly discovered user defined data type [3]. We propose an alternative solution by introducing a pattern language for generic description of the Clean-CORBA skeletons. The skeletons are extended with the description of the formal parameters: type parameters, strategy etc. The user of the skeletons provides the description of the actual parameters corresponding to the formal parameters. Also the interface of the CORBA server objects referenced by the parameterized clients are described in a generalized way. Additionally to the Clean and the IDL compilers we propose a code generator, which interprets the formal and actual parameters, generates IDL descriptions and instances of the server objects according to the actual parameters, invokes the IDL and the Clean compilers for generating stubs and objects.

The second problem is solved by running objects at each computing node, which objects starts the components of the skeleton according to the actual strategy.

The third problem can be solved by upgrading the current Clean-CORBA interface to the Clean 2.1 compiler and Object IO library which supports threads, event streams, callback mechanism and asynchronous communication.

Categories and Subject Descriptors: D.1 Programming Techniques: D.1.1 Applicative (Functional) Programming, D.1.3. Concurrent Programming.

Key Words and Phrases: Skeleton, distributed functional programming, Clean, CORBA, design patterns, middleware.

*Supported by the Hungarian National Science Research Grant (OTKA), Grant Nr.T037742.

References

[1] Horváth Z., Zsók V., Serrarens, P., Plasmeijer, R.: Parallel Elementwise Processable Functions in Concurrent Clean, Mathematical and Computer Modelling 38, 2003, pp. 865-875.

[2] Zsók V., Horváth Z., Varga Z.: Functional Programs on Clusters In: Striegnitz, Jörg; Davis, Kei (Eds.): Proceedings of the Workshop on Parallel/High-Performance Object-Oriented Scientific Computing (POOSC'03), Technical Report, FZJ-ZAM-IB-2003-09, July 2003, pp. 93-100.

[3] Mowbray, T.J., Malveau, R. C.: Corba Design Patterns, Wiley Computer Publishing, 1997.