Partially Distributed Objects

Markus Hof

Institute of Computer Science (System Software)
Johannes Kepler University
Altenbergerstr. 70
A-4040 Linz

Tel.: +(43)+(732) 2468 7132
Fax: +(43)+(732) 2468 7138
Internet: hof@ssw.uni-linz.ac.at

Abstract. We introduce light weight distributed objects. The main goal is to ease the transition of existing object-oriented applications, not aware of distributed available objects, to applications which do support them. This implies for transparent method invocations on distributed objects, i.e. methods may have arbitrary parameters. Pointers are handled either through a deep copy, or a shallow copy which keeps the identity of the copied object. Stub and proxy code is only generated on demand (just-in-time code generation). Especially, our distributed objects support distributed models, to allow applications written using the MVC concept, to offer their models over the network. We introduce a notification mechanism from server object to surrogate object, partially distributed method invocation, and just-in-time generation of stub code. Our objects build upon a generic network interface which allows access over arbitrary networks (LANs, Internet, ...).

Oberon is a single-user operating system running either native, or on top of most available computer systems. It has its own user interface; it is written in the object-oriented programming language Oberon-2; its user interface is built using the MVC (Model-View-Controller) concept. Each model (file, text, spreadsheet, drawing, ...) may be simultaneously visible in several - possibly even different - views.

The purpose of this project is to add support for distributed objects to the Oberon system. In particular the distributed objects should support the implementation of distributed models, i.e. they should support the adaptation of existing Oberon applications (models) to a distributed environment. To ease this transformation we proclaim the following concepts: Automatic just-in-time generation of stub code, partially distributed method invocation, efficient notification from a server object to its surrogate objects.

In order to add distributed objects, we had to do some preliminary work. First, the Oberon system implements only cooperative multitasking, which allows a task to