

発表概要

使いやすさ・透過性・拡張性の高い分散オブジェクトシステム RyORB

根 山 亮[†] 首 藤 一 幸[†] 村 岡 洋 一[†]

本研究では、より使いやすく、透過性・拡張性の高い ORB RyORB を Java 言語上に実装した。RyORB では、既存のクラスのオブジェクトの遠隔オブジェクト化や、さまざまな種類の遠隔メソッド呼出しを実現した。実験の結果、従来の JavaORB と比較して遜色ない遠隔メソッド呼出し性能が得られた。

従来の JavaORB では、メソッド呼出しのインターフェース定義が必須であったり (RMI)、遠隔オブジェクトへの参照であるスタブの遠隔オブジェクト型の変数への代入やメソッドへの引数渡しのような透過的な扱いができなかった (HORB, Voyager)。また、スタブクラスの設計が引数渡しと返り値の受け取りを同時に行うという特定の呼出しに特化しているため、非同期呼出しのような新たなメソッド呼出しの実現には、生成するスタブクラスのコードを変更する必要があった。

RyORB では開発の利便性を考慮し、スタブクラスを (1) 遠隔オブジェクトクラスのサブクラスとする方法と (2) 遠隔オブジェクトクラスと同一のインターフェースを実装したクラスとする方法を、開発者が選択できるようにした。また、拡張性を考慮しスタブクラスを汎用化したことにより、スタブクラスを一切変更せずに様々な呼出しを容易に実現できた。実際に UDP による呼出し、返り値を待たない呼出し、非同期呼出し、UDP のブロードキャストで複数の遠隔オブジェクトからの返り値を受け取る呼出しが可能となった。

RyORB: An Easy-to-Use, Network-Transparent, and Extensible Distributed Object System

RYO NEYAMA,[†] KAZUYUKI SHUDO[†] and YOICHI MURAOKA[†]

This paper presents a Java ORB which is easy to use for developers, more transparent, extensible to various types of remote method call, called RyORB. It enables to change instances of existing classes into remote objects, and enables various types of remote call. RyORB showed good performance for experiments of remote call.

In most previous Java ORBs, interface definition was forced (RMI), it was impossible to handle remote objects transparently, like substituting a stub for a variable declared as the remote object class (HORB, Voyager). Also, stub codes must be modified to add a new type of remote call, because their design of stubs was specialized to a specific type of remote call, like sending parameters and receiving a return value simultaneously.

In RyORB, developers can select a method of implementing a stub from the two methods: (1) implementing as a subclass of the remote object class, (2) implementing as a class which has the same interface as the remote object class has. Furthermore, the design of stub classes is flexible and extensible, so various types of remote call are enabled easily without any modification of stub codes such as calls via UDP, oneway calls, future calls, and calls by UDP broadcast, in which a caller receives some return values from remote objects.

(平成 11 年 3 月 24 日発表)

[†] 早稲田大学理工学部
School of Science and Engineering, Waseda University