

---

**発表概要**

---

## 分散透過デバッガ機能の拡張

高木利公<sup>†,††</sup> 中里秀則<sup>††</sup>

複数の計算機上に分散した、分散オブジェクトシステムを用いて通信するサーバ/クライアントモデルのようなプロセス群で構成される分散アプリケーションの開発（設計、製造（コーディング）、試験）を容易にするために、CORBA や Java 等の技術が開発されている。アプリケーションの開発者は、分散アプリケーションを設計/製造するにあたり、プロセス間通信に CORBA を用いて分散を意識せずに設計/製造ができる。しかし試験時は、複数の計算機上の分散したプロセス群からバグを持つと思われるプロセスを切り分けながらデバッグしなければならない等、分散に起因する困難さがある。我々はすでにこの問題を解決する手法として、複数のプロセスを 1 つのデバッガシステムからデバッグ可能な「分散透過デバッガ」を提案し、実装の報告をした。ここではすべてのプロセスを同時にソースコードデバッグできる等の成果を得たが、「すべてのプロセス群が正常に規定したインタフェースで動いていること」を確認するといった大規模プロセス群の統合試験時は、利便性から問題が残った。この観点から、我々は分散透過デバッガにプロセス間通信のすべてのメッセージを捕捉し、ブレイクポイントを設定できるメッセージデバッガ機能を組み込み、オープンな統合分散透過デバッガの設計・実装を行った。本発表では、このアプリケーションのソースコードを変更することなく適用できるメッセージデバッガ機能の構成、言語透過性を提供する仕組み等について説明する。

### Extension of Distribution Transparent Debugger

TOSHIHIRO TAKAGI<sup>†,††</sup> and HIDENORI NAKAZATO<sup>††</sup>

CORBA and Java simplify design, coding, and test of distributed applications. Distributed applications are composed of a group of processes. The processes may communicate with each other in client-server model provided by distributed object systems. Application developers do not need to be aware of distribution by using CORBA for inter-process communication when they design and code the applications. However, when they test the applications, they suffer from the difficulty caused by distribution. For example, the developers must find the process that may contain a bug among processes that are distributed over many computers. We have proposed and implemented “distribution transparent debugger” to overcome this problem. The distribution transparent debugger provides source-code debugging of all involving processes from a single debugger. Although the distribution transparent debugger solved some of the problems in distributed application test, we still have problems such as confirming all processes are operating with the right interfaces in integrated test over large number of processes. We developed the message debug function that capture inter-process communication messages and set break points on messages. In addition, we made our debugging environment transparent to programming languages. In this presentation, we present the structure of the message debug function and the mechanism to provide language transparency.

（平成 15 年 3 月 18 日発表）

---

† 大学院国際情報通信研究科  
Graduate School of Global Information and Telecommunication

†† 早稲田大学  
Waseda University