

発表概要

プログラムの部分移送に基づく遠隔実行機構とその知的インタフェースへの応用

倉林 則之[†] 湯浅 太一^{††} 小宮 常康^{††}

本稿では、Java VM上で動作するScheme処理系「ぶぶ」における、遠隔実行機構の導入とその応用について述べる。プログラムの部分移送をとまなう遠隔実行は、ソケット通信や遠隔手続き呼び出しとともに開放型分散環境におけるシステムプログラミングをサポートするための言語機構として利用できるが、ここでは特に、プログラムの移動が本質的な役割を果たす、移動可能な知的インタフェースに焦点を当てて、遠隔実行の応用について議論する。我々の遠隔実行機構は、指定された部分を本体とする関数閉包のコピーによって実現されている。遠隔実行は同期型で行われるが、この間、接続は維持する必要が無い。関数閉包の環境は実行結果とともに呼び出し側に返される。このため、必要な部分のみを効率的に移送できるだけでなく、通信環境の変化に対しても頑強である。具体的な応用例として、URLを選択してWebブラウザを起動するアプリケーションを作成した。このアプリケーションは、閲覧したURLと時刻を事例として学習し、過去の最も近い時刻に閲覧されたURLを候補として推薦する、予測インタフェースを実現している。事例はサーバに蓄積され、GUIを実装しているクライアント上で学習機構が遠隔実行される。新たに追加された事例をサーバに保存することで、クライアント間で学習結果を共有することが可能となる。このアプリケーションの実装を通して、我々の遠隔実行機構の有効性を示す。

Remote Execution by Partial Program Transfer and its Application to Intelligent Interfaces

NORIYUKI KURABAYASHI,[†] TAIICHI YUASA^{††}
and TSUNEYASU KOMIYA^{††}

This paper describes the remote execution mechanism on Bubu, a Scheme system running on Java VM, and its application to intelligent interfaces. Remote execution by program transfer is a generic language mechanism which supports system programming in open distributed environments. In this paper, we discuss applications of remote execution, focusing on intelligent interfaces for which program migration plays an essential role. Our remote execution mechanism is implemented by copying a function closure that contains the specified part of the program to be transferred. Although communication is always synchronous, the connection need not be maintained during remote execution. The environment enclosed in the closure will be returned to the caller, together with the execution result. Accordingly, the partial program can be transferred effectively and the execution is robust against changes of network environment. As a concrete example of applications, we implemented a program for launching a Web browser which automatically selects an appropriate URL. This program realizes a predictive interface which learns the accessed URLs and their time-stamps as the user profile, and recommends an appropriate URL at the startup. The URL recommended is determined by comparing time-stamps of the URL/time-stamp pairs. The user profile is stored on the server, and the learning engine is executed remotely on the client that implements the GUI. Since newly acquired profile is appended to the user profile on the server, clients can share the profile. We examine the implementation of this application, and show the effectiveness of our remote execution mechanism.

(平成 12 年 3 月 24 日発表)

[†] エイ・ティ・アール環境適応通信研究所

ATR Adaptive Communications Research Labs.

^{††} 京都大学大学院情報学研究科

Graduate School of Informatics, Kyoto University