6J－11

Javaによるリモートプログラムのデバッグ手法の提案*

片山　透　中本　幸一　臼井　和敏†
NECソフトウェアデザイン研究所組込ソフトウェア技術部‡

1. はじめに

一般に分散環境でのリモートプログラムのデバッグは，デバッガとデバッグ対象がそれぞれ異なる環境で実行されるため，困難であることが知られている．その解決策としてリモート環境にデバッガを設置したり，デバッグのためのライブラリを用意するなどの手法が提案されている[1][2].

本稿では，特にJavaのリモートプログラムに関して，そのリモート環境に依存しないデバッグのためのフレームワークと，それに基づくデバッグ機能の実装例について紹介する．

2. Javaのリモートプログラム

Javaはリモートプログラムの記述や実行のための機能を備えており，さまざまな形態のリモートプログラムが構築可能である．本稿では，その形態を以下のように分類する．

- メッセージだけが伝播される
  RMIプログラム，ソケットプログラム
- クラスファイルが移動する
  アプレットプログラム
- オブジェクトが移動する
  RMIプログラム，エージェントプログラム

これら，Javaのリモートプログラムをデバッグするための手法は，従来のリモートプログラムのデバッグ手法と同様の方法が考えられる．しかし，この方法ではリモート環境に依存するという問題がある．

Javaには実行環境に依存しないという特徴があり，デバッグにおいても任意の実行環境を想定する必要がある．しかし，従来手法では，こうした環境には対応できない．

3. Remote Debug Frameworkの構成と特徴

本稿では，Remote Debug Framework(以下，RDFと省略する)はJavaによるリモートプログラムをデバッグするためのフレームワークを提案する．

3.1. RDFの構成

RDFの構成を以下の図1に示す．

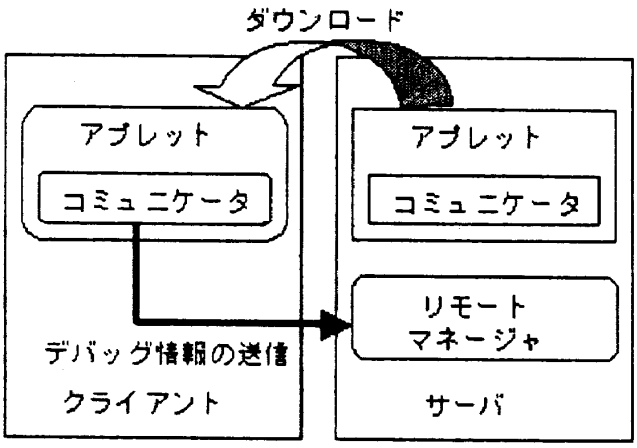

図1 RDFの構成

RDFは図1に示す通り，大きく次の2つのモジュールで構成されている．

- コミュニケータ
  デバッグ対象のアプレットと一緒に移動して，クライアント上でデバッグ情報を収集して，サーバに送信したり，サーバからのコマンドを実行したりする．
- リモートマネージャ
  サーバ上で動作し，コミュニケータに対してデバッグコマンドを送信したり，コミュニケータから送信されるデバッグ情報をユーザインタフェースに表示したり，サーバ上に保存する．

---

* Proposal of debug method for remote program by Java.
† Toru Katayama, Yukikazu Nakamoto, Kazutoshi Usui.
‡ NEC Software Design Laboratories.

3.2.RDF の特徴

RDF の特徴を以下に挙げる．

- 任意のリモート環境でデバッグ可能

  デバッグ情報を収集するコミュニケータがデバッグ対象のアプレットと一緒に移動するため，リモート実行環境に依存しない．

- さまざまなデバッグ機能を実装可能

  デバッグ機能はインタフェースを実装することでさまざまな機能を実現できる．

このように RDF は，リモート実行環境に依存しないという特徴を備えており，また，さまざまなデバッグ機能を容易に実現することを可能とする．

4. デバッグ機能の実装例

RDF のデバッグ機能の一実装例として Remote ThreadViewer(以下，RTV と省略する)を紹介する．

ThreadViewer とは，Java プログラムのスレッドの振る舞いを視覚化するデバッガであり，時間軸上でスレッドの切り替わりを観察することでデッドロックなどのマルチスレッドプログラムのデバッグ支援が可能となる[3]．RTV はその機能を RDF 上で実現したものである．

RTV では，コミュニケータはデバッグ対象となるアプレットプログラムのスレッド情報を収集して，サーバ上のリモートマネージャに送信する．リモートマネージャは送信されたスレッド情報を基に**図 2** のようにスレッドの振る舞いを表示する．

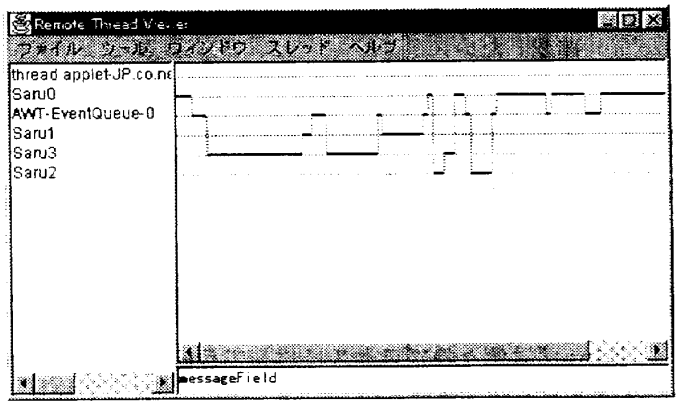

図 2 RTV での表示例

RTV を用いることにより，任意のリモート環境で実行中のアプレットプログラムの振る舞いを観測することが可能となる．

5. RDF の問題点

RDF を用いていくつかのデバッグ機能を実装したところ，以下のような問題があることがわかった．

- デバッグ機能の限界

  デバッグ機能を Java で実装しているので、実現する機能が Java の能力により制限される．例えば，通常のシンボリックデバッガなどのような詳細な実行制御を実現するのが困難であるなどの問題がある．

- セキュリティ上の制限

  アプレットプログラムがデバッグ対象であるため，セキュリティ上の制限で必要な情報が得られない場合がある．例えば，リフレクション機能を利用した場合，リモート側のクラスにはアクセスできないなどの問題がある．

これらの問題は，リモート環境の JavaVM を特定したり，ブラウザのセキュリティを解除することで解決可能である．しかし，この条件はリモート環境の JavaVM を特定しないという RDF の前提に反するため，現状ではこれらの問題の範囲内でデバッグ機能の実現を目指すこととした．

6. おわりに

本稿では RDF のデバッグ対象として，アプレットプログラムを挙げたが，RMI プログラムなど，他の Java によるリモートプログラムへの適用も今後検討していく計画である．

参考文献

[1] 植木他，"組み込み型分散システム用デバッガ"，第 46 回情報処理全国大会論文集 7E-9，1993.

[2] 横山他，"並列処理プログラム用リモートデバッガ"，第 51 回情報処理全国大会論文集，1L-5，1995.

[3] 片山他，"Java プログラムのスレッド視覚化ツール"，情報処理学会研究報告 97-SE-115-6，1997.