

発表概要

実機における Java アプレットのモニタリング方式の提案

吉田 聡[†] 大原 茂之^{††}

Java アプレットを利用する携帯情報端末や組み込み機器などの実機テストあるいはフィールドでのメンテナンスを行う場合、その Java アプレットが正しく動作しているかどうかをリモートで検証しなければならないという問題がある。アプレットの動作を検証するには、実機におけるイベントの発生や処理のタイミング、およびプログラム間のメッセージの送信内容といった情報を取得する必要がある。しかしながら、現状のモニタリングツールで Java プログラムの動作情報を取得しようとすると、メッセージ送信などを追加するために、すでに作成済みのメソッドを直接修正しなければならない。このため、動作情報を取得するためのアプレットと実際に動作するアプレットが異なったものとなり、テスト段階から実用段階へ移行する際にバグが発生する恐れがある。本発表では、Java プログラムのメソッドを変更することなく、実機におけるプログラムの動作をモニタリングする方法について提案する。さらに、モニタリングした情報をオブジェクト図上でビジュアルに表示する環境についても提案する。ここで提案する開発環境は、既存の Java プログラムを入力すると、そのプログラムのメソッドをブラックボックス化してモニタリング用プログラムに組み込む。このプログラムを実行させると、イベントやメッセージの履歴をモニタリング結果として保持する。さらに、モニタリング結果をオブジェクト図上でアニメーション表示させる。これによって、Java プログラムのシミュレーションやモニタリングから実行へ移行する際におけるバグの発生を防ぐことができるようになる。また、実行結果に誤りがある場合には実機を用いてメンテナンスすることも可能となる。

Monitoring Method of Java Applet on Target Machine

SATOSHI YOSHIDA[†] and SHIGEYUKI OHARA^{††}

When we perform target machine test or maintenance for Personal Digital Assistant or Embedded Systems which use a Java applet, there is a problem that we must verify by remote whether the Java applet is operating correctly. For verifying operation of an applet, it is necessary to acquire information of timing of generating or process of an event, contents of communication of a message. However, if we are going to acquire the operation information on Java program with a present monitoring tool, we have to correct a program's method directly, in order to add message transmission. For this reason, it becomes the applet for acquiring operation information and the applet that actually operates are different. Therefore, when shifting to a practical use stage from a test stage, there is a possibility that a bug may occur. In this presentation, we propose about the monitoring method of the operation of the program in target machine, without changing the method of Java program. Furthermore, we propose also about the environment that displays visually the information that carried out monitoring on an object figure. By using this method, we can protect generating of the bug, when we shift to a practical use stage from the test stage of Java program. Moreover, supposing an error is in an execution result, we will become possible also maintaining using target machine.

(平成 14 年 6 月 17 日発表)

[†] 愛知学院大学商学部産業情報学科

Department of Business Information, Faculty of Commerce, Aichi Gakuin University

^{††} 東海大学電子情報学部情報メディア学科

Department of Information Media Technology, Faculty of Electronics Information Technology, Tokai University