

JAVA アプレットリモートメンテナンス技術について

6H-7

高橋 浩也[†] 吉田 聡^{††} 大原 茂之[†]

[†]東海大学 ^{††}愛知学院大学

1 はじめに

リモートによってダウンロードされた JAVA アプレットのメンテナンスやデバッグをする上で、クライアント側における JAVA アプレットの動作をサーバ側において検証することは困難であった。

本報告では、クライアント側における JAVA アプレットの動作情報をサーバ側において取得し（以下、モニタリングと記す）、その情報をビジュアルに表示することによってメンテナンス等に役立てる方法について提案する。

2 JAVA アプレットのリモートメンテナンス

JAVA アプレットリモートメンテナンスとは、JAVA アプレットをメンテナンスする際に、キーボード入力、マウスクリックおよびプログラム間のメッセージ交信などのイベントをモニタリングし、メンテナンス等に役立てることである。

JAVA アプレット上におけるイベントを、JAVA アプレット実行情報と呼ぶ。JAVA アプレット実行情報は、イベントが発生するたびに、Web サーバへ送られる。ユーザは、Web サーバへ送られた JAVA アプレット実行情報をもとにメンテナンス等を行う。

3 JAVA アプレットモニタリングシステム

3.1 JAVA アプレットにおける

イベントのモニタリング

JAVA アプレットにおけるイベントをモニタリングするためのデータの流れは図 1 のようになる。図 1 において、B_Applet は作成された JAVA アプレットを保持するプログラムである。B_Applet はユーザからのイベントを受け、そのイベントを作成された JAVA アプレットに渡す機能を持つ。

JAVA Applet Remote Maintenance Technology.

Hiroya TAKAHASHI[†], Satoshi YOSHIDA^{††},

Shigeyuki OHARA[†].

[†]Tokai University, ^{††}Aichi Gakuin University.

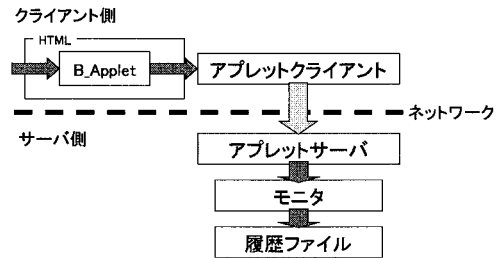


図 1 JAVA アプレットにおけるイベントをモニタリングするためのデータの流れ

図 2 に JAVA アプレットのイベントの受け渡しを示す。図 2 において、B_Applet は JAVA アプレットを保持するプログラムである。また、B_Ap は JAVA アプレットと同一のイベント(E1)を持つ。ユーザからのイベント E1 が B_Applet に入力されると、B_Applet は、E1 を JAVA アプレットに渡すと同時に、アプレットクライアントへ JAVA アプレット実行情報を送信する。また、E1 に対する JAVA アプレットの反応が B_Applet に渡されると、B_Applet は JAVA アプレットの反応を出力すると同時に、アプレットクライアントへ JAVA アプレット実行情報を送信する。

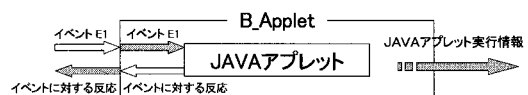


図 2 JAVA アプレットのイベントの受け渡し

アプレットクライアントは、B_Applet から JAVA アプレット実行情報を受けると、ネットワークを通してアプレットサーバに JAVA アプレット実行情報を送る。アプレットサーバは、アプレットクライアントからの JAVA アプレット実行情報を受けると、モニタオブジェクトへ JAVA アプレット実行情報を送る。モニタは、アプレットクライアントから受けた JAVA アプレット実行情報をもとに、ダウンロード先の IP アドレス、JAVA アプレットにおけるイベントをモニタリング結果として履歴ファイルに記録する。また、モニタリン

グ結果をもとにダウンロード先のアプレットの動作を Web サーバ上で再現する。

3.2 プログラム間におけるメッセージの受け渡しとオブジェクトの保持

JAVA プログラムから送信されるメッセージは、JAVA 仮想機械によって送信先のプログラムに対して直接送信される。また、メッセージは送信先以外のプログラムに送信されることはない。このため、JAVA プログラム間でのメッセージの交信が正しくなされているかをモニタすることは困難であった。そこで、JAVA アプレットモニタリングシステムは送信先のプログラムと同じ名前のプログラムを用意し、そこにメッセージを送信させるようにする。図 3 にプログラム間におけるメッセージの受け渡しとオブジェクトの保持を示す。

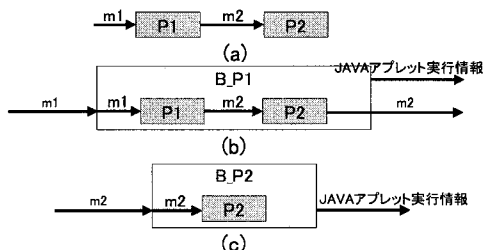


図 3 プログラム間におけるメッセージの受け渡し

図 3(a)は、JAVA アプレットにおける既存のプログラム P1 と P2 である。P1 は入力メッセージ m1 と出力メッセージ m2 を持つ。(b)は、P1 を保持する B_P1 である。B_P1 が保持する P2 は、JAVA アプレットモニタリングシステムによって作成されたものであり、P1 から m2 を受信するとモニタに対して JAVA アプレット実行情報を送信する。B_P1 に保持されている P1 が P2 にメッセージ m2 を送信したとき、m2 を受信するのは B_P1 に保持されている P2 である。(c)は、P2 を保持する B_P2 である。

4 JAVA アプレットモニタリングシステムの適用例

電卓を例として、JAVA アプレットリモートメンテナンスについて説明する。図 4 に電卓のオブジェクト図を示す。

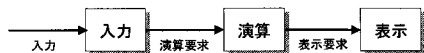


図 4 電卓のオブジェクト図

図 4 において、ボタン入力、ユーザからのボタン入力を受け、演算要求およびボタン入力情報を演算へ渡す。演算は、ボタン入力からの演算要求およびボタン入力情報を受けると、表示要求および演算結果を表示へ渡す。表示は、表示要求および演算結果を受けると、画面表示を行う。図 5 にシステムの実行例を示す。

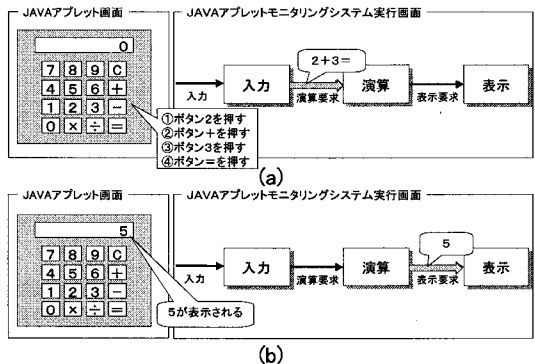


図 5 システムの実行例

図 5 (a)において、ユーザは、JAVA アプレット上に表示された電卓のボタンを”2”、”+”、”3”、”=”の順に押す。サーバ側では、ボタン”2”、”+”、”3”、”=”が押されると、それに対応して、入力から演算への演算要求メッセージが強調して表示され、データ”2+3=”が吹き出しで表示される。同様に(b)では、JAVA アプレット上のテキストフィールドに 5 が表示されると、それに対応して、演算から表示への表示要求メッセージが強調して表示され、データ”5”が吹き出しで表示される。

このように、JAVA アプレット上のイベントに対応したオブジェクト間のメッセージ交信を、図で動的に検証することができる。

5 おわりに

本報告では、JAVA アプレット上のイベントをモニタリングし、JAVA アプレットの動作をビジュアルに表示、検証を行う方法を提案した。これにより、JAVA アプレットのメンテナンスが容易になる。今後は JAVA アプレットモニタリングシステムの実装、評価を行う予定である。

参考文献

- 1) 浦野, 大原, 村松, 長谷川: “システム開発における要求仕様のビジュアル化に関する研究”, 情報処理学会 58 回全国大会