単一の Java Applet で構築する複数画面のシステム

アプレットサーバシステム研究所　柳瀬　隆敏

１．はじめに

GUI ウィンドウの描画には、大きく分けて２種類の方法がある。(i)ウィンドウ全体を描画する方法と、(ii)ウィンドウ内の指定された表示要素だけを描画する方法である。

(i)は、新規にウィンドウを表示するときのほか、重なり合ったウィンドウを描画するときに使われる。ウィンドウの重なり具合に変化が生じると、重なり合ったウィンドウについて、他のウィンドウに覆われているウィンドウから順に（つまり、下にあるウィンドウから順に)、ウィンドウマネージャが描画指示を送る。ウィンドウマネージャから描画指示を受けたウィンドウは、ウィンドウ全体のイメージデータをグラフィックメモリに上書きする。下にあるウィンドウが、後からイメージデータをグラフィックメモリに上書きすると、下にあって隠れているはずの部分が表示されてしまう。したがって、ウィンドウマネージャの指示なしに、ウィンドウがウィンドウ全体の描画をすることはできない。

通常、ウィンドウは、(ii)の方法で描画する。CardLayout あるいはタブ画面という GUI 部品がある。これは、同じ大きさのカード状の表示要素を複数個重ねてウィンドウ上に配置しておき、表示したいカード状の表示要素を一番上に移動させ、(ii)の方法で描画するものである。カード状の表示要素だけを表示する方法で、画面を切替えている。この方法では、切替える画面の数が増えると、メモリの使用量が増える。

２．概要

（１）ウィンドウの切替え表示

GUI ウィンドウを持つクライアントサーバシステムが多用された時期がある。このようなシステムでは、子ウィンドウを生成し、異なる画面を表示していた。一旦表示されたウィンドウを、別のウィンドウに切替え表示できなかったからである。前述のタブ画面のような方法で表示を切替えることはできる。しかし、切替えるタブ画面が増えるとメモリの使用量が増える。

本技術では、ウィンドウ全体のイメージデータを、(i)の描画方法を使ってグラフィックメモリに上書きさせる。イベント処理プログラムで、①GUI 部品を削除後、同じイベント処理プログラムで新しい画面の GUI 部品を貼り付ける。さらに、②同じイベント処理プログラムでウィンドウの大きさを利用者に判らない程度で変える。ウィンドウの大きさが変わると、ウィンドウの重なり具合が変わるため、ウィンドウマネージャが、ウィンドウに描画指示を送る。ウィンドウマネージャから描画指示を受け取ったウィンドウは、ウィンドウ全体のイメージデータをグラフィックメモリに上書きしウィンドウ全体を描画する。これで、GUI ウィンドウを切替え表示できる。

Java の GUI である Swing では、①の処理を行っただけでは、グラフィックメモリにイメージデータが上書きされない。そのため、②の処理を行ってウィンドウ全体を描画させている。

上記技術を Java Applet に適用すると、単一の Java Applet で複数画面のシステムを構築できる。従来技術である WEB アプリより単純な仕組みで画面切替えができる。

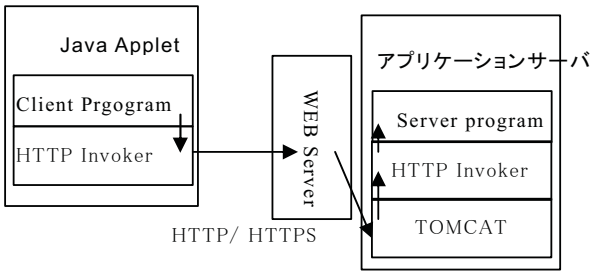
（２）Java Apple とアプリケーションサーバの通信

SpringFramework というフレームワークには、HTTPInvoker という、リモートプロセジャコールの仕組みがある。HTTPInvoker を使うことによって、Java Applet とアプリケーションサーバ間でリモートプロセジャコールができる。Java Applet と WEB サーバまでの間は、HTTPS で暗号化可能、WEB サーバとアプリケーションサーバ間の暗号化は任意である。次のプログラムは、HTTPInvoker を使ったリモートプロセジャコールの例である。

```
ds = (DoSomethingIF)factory.getObject();
ds.exec(・・・・・・・);
```

下の図は、HTTPInvoker の概念図である。

HTTP Invoker の概念図



この概念図を使って、プログラム例の２行目の HTTPInvoker の処理の流れを説明する。

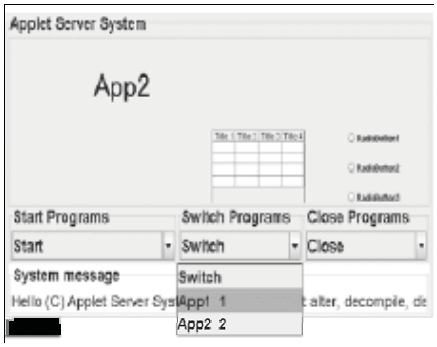
Java Applet の Client Program から端末の HTTPInvoker に対してプロセジャコールを行う。端末の HTTPInvoker は、受け取ったプロセジャコールを HTTP または HTTPS で WEB サーバに送信する。WEB サーバは、受け取ったプロセジャコールをサーブレットサーバの TOMCAT に転送する。TOMCAT では、サーバ側の HTTPInvoker が動作していて、受け取ったプロセ

ジャコールが、サーバ側の HTTPInvoker に渡される。サーバ側の HTTPInvoker は、受け取ったプロセジャコールに基づき、Server Program を実行する。実行結果は、逆の経路で Java Applet の Client Program に送られる。

３． 試作例

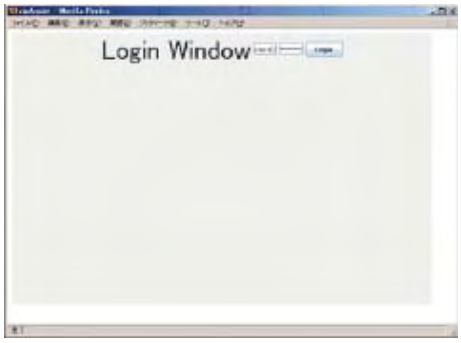
以下のようなシステムを試作した。

（１）１つ Java Applet で複数のプログラムを切替え表示する例である。画面左下の Start メニューでプログラムを起動する。中央下の Switch メニューで実行中のプログラムを切替える。右下の Close メニューでプログラムを終了する。

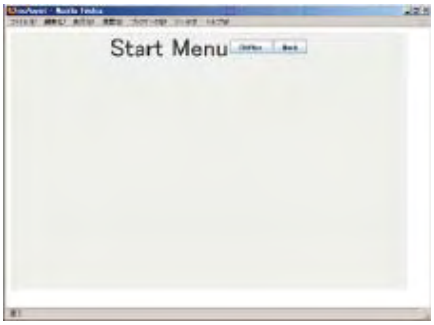


（２）１つの Java Applet で、画面切替えしたあと別ウィンドウでプログラムを起動する例である。すべてのウィンドウが単一の Java Applet で表示される。

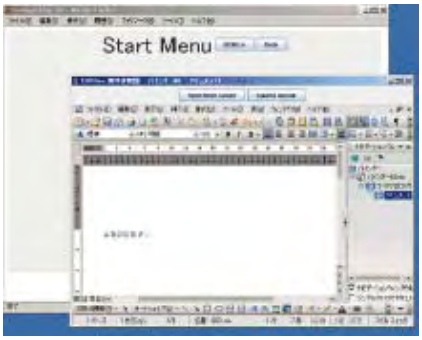
次の図のように、１つの Java Applet の中で、ログイン画面　→　メニュー画面　と切替え表示したあと、別ウィンドウでプログラムを起動する。別ウィンドウのプログラムも Java Applet の一部として動作する。この例では、市販されている Java １００％の Office suite EIOffice を実行している。EIOffice を Java Applet の一部として動作させることにより、HTTPInvoker を使ったリモートプロセジャコールで、サーバにもデータを保管できる。



↓　メニュー画面に切替え表示



↓　新規ウィンドウでアプリを実行



４． おわりに

Java Applet は、元祖クラウドコンピューティングと言える。当初は、動作が遅いなどの使いにくさがあった。Java 100%の EIOffice を使ってみれば判るように、動作速度は実用上問題ない程度に速くなっている。本稿の技術を使えば、複数の画面を切替えるシステムが簡単に構築できる。したがって、GUI を使う業務システムなどの基盤技術として、Java Applet が１つの有力な選択肢になる可能性がある。

文　献

河村義之　首藤智大　竹内祐介　吉尾真祐　著　実践 Spring Framework 日経 BP 社 2005

宮本信二　飯田伸一　青木祐　著 Eclipse 導入ガイド ソフトバンク　パブリッシング 2003