

B-29

ASPによる統合リテラシツール

An Integrated Tool for Computer Science Literacy by ASP

高橋 栄† 森 秀樹† 上原 稔†

Sakae Takahashi Hideki Mori Minoru Uehara

E-mail : gz0200101@toyonet.toyo.ac.jp

1. まえがき

Windows 98 が登場した頃から、コンピュータに興味を持たない人も「インターネット」という言葉を耳にするようになった。

最近のコンピュータには、Web ブラウザや電子メールソフトが標準で用意されている。そのほかにもインターネットを利用した様々なサービスがある。その中に最近現れた ASP というサービスがある。ASP とはアプリケーションをネットワーク経由で提供するサービスである。これによりインストールという作業をせずにアプリケーションを利用できるのである。

もちろん利用者が、アプリケーションを自分のコンピュータにインストールしてしまえばそれでよい。多くの場合、今まではそのようにしてコンピュータは利用されてきた。そして「インターネット」が広まると同時にコンピュータ利用者が増え、利用者の知識や経験の幅も大きく広がった。そのため、必ずしも扱いに慣れた者だけがコンピュータを利用するとは限らなくなった。それにしても初心者にとって、インストールというのは敬遠したい作業である。説明文中の用語の意味がわからなかったり、手順に不明な点があったり、不安を感じながら行う作業がほとんどだからである。

仮にインストールできたとしても最近のアプリケーションは非常に多機能である。初心者はまずその機能の多さに圧倒されるであろう。どのようなことができるのか、どこをどうすれば目的の機能が利用できるのか検討もつかないのでないだろうか。

しかしながら世に氾濫する文書のほぼ全てがコンピュータで作成されたものである。もっとも身近な文書である「手紙」ですら「電子メール」に置き換えられつつある。コンピュータでの文書作成はもはや常識であると言えるだろう。それに伴い、新規のコンピュータ利用者を対象にした手軽な文書作成の学習ができるアプリケーションが必要なのではないだろうか。

本研究では統合リテラシツール (エディタ、ワープロ、HP 作製ソフト等) を ASP (Application Service Provider) で提供するシステムを提案する。本システムは Java アプレットであるため「利用するプラットフォームは限定されない」「煩わしいアプリケーションのインストール作業を必要としない」これによりインストールの問題は解消される。更にリモートファイルサーバを設置し、そこに作成されたデータを保存させる。従って、Java アプレットが利用できる環境であれば、いかなる端末からでも作業の続行が可能である。

本論文では提案システムの設計、実装について述べる。

2. RMI (Remote Method Invocation)

本研究では Java 言語で開発するほか、ファイルサーバからリモートでデータを扱えるようにするために RMI [1][2] を用いる。その利点として

・ 変更に対する柔軟性

RMI では、サーバとクライアントは Java のインタフェースで結合される。これはインタフェースが変更されない限り、サーバの処理に変更があっても、クライアントを変更する必要がなく、また、サーバの変更無しでクライアントを変更できることを意味する。したがって RMI では、このようにサーバ、クライアントの変更に対し柔軟なシステムを構築できる。

・ 記述の容易性

RMI を用いてサーバ、クライアント・プログラムを記述することは簡単である。サーバ・プログラムでは、それが RMI のサーバであることを宣言するための数行のコードを記述するだけである。

・ 可搬性

RMI は、JDK1.1 以降でコアパッケージとして提供されている。従って、RMI を使用して作成されたプログラムは、JDK1.1 以降の環境があれば、他のどのようなシステムにおいても動作させることができる。

・ データ送受信の容易性

RMI ではメソッドを呼び出す際の引数、戻り値として、オブジェクトを受け渡すことができる。したがって、ハッシュテーブルのような複雑なデータ構造でも、ただ一つの引数として取り扱うことができる。

・ 並列処理

RMI はマルチスレッド対応であり、複数のクライアントの要求は並列に処理される。

などが挙げられる。

本研究では特にデータ送受信の容易性に着目した。

RMI を用いずソケットを用いた場合、どこのサーバの何番のポートに接続するか等のプロトコルを記述しなければならないなど RMI を用いた場合と比較して手間がかかるからである。

3. RMI のクラスとインタフェース

リモートオブジェクトの作成の際、`java.RMI` , `java.RMI.server` パッケージのクラスを用いた。

- Remote インタフェース

RMI の中でリモートオブジェクトやスタブ等を Remote 型のオブジェクトとして扱うために用いられる。

- RemoteObject クラス

`hashCode` , `equals` , `toString` メソッドをオーバーライドすることにより `Object` クラスが提供している機能をリモートオブジェクトに対して利用できるように拡張したクラスである。

- RemoteServer クラス

全てのリモートオブジェクトの共通のスーパークラスである。

- UnicastRemoteObject クラス

Point-to-point のストリームを使ったリモートオブジェクトを実現するためのクラスである。このクラスのサブクラスとして新たにリモートオブジェクトを作成すると、コンストラクタにより、リモートオブジェクトが他のマシンから自動的に参照可能になる。

4. システムの設計

4.1 アーキテクチャ

スタブ及びスケルトンはコンパイル時に自動的に生成される。RMI を用いたときのアーキテクチャ図 1 に示す。

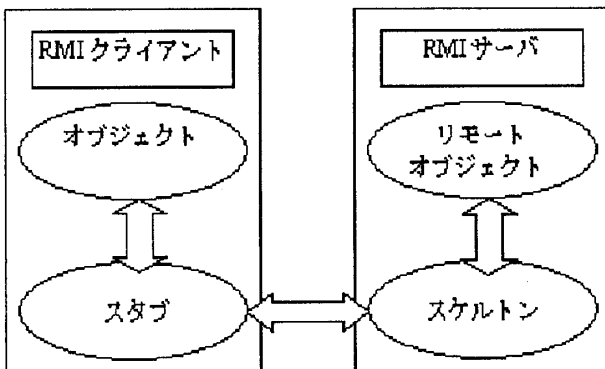


図 1 : RMI のアーキテクチャ

クライアントオブジェクトとリモートオブジェクトの通信は、スタブとスケルトンを介して行われる。

スケルトンでははじめにスタブから書き出された引数をストリームから読み込み、リモートオブジェクトのメソッドを実行し、結果をスタブへのストリームに書き出す。

スタブはストリームを使用して引数をスケルトンに送信し、戻り値をストリームから取り出す。

4.2 クラスの構成

NotePad クラス

テキストエディタ本体のクラス。

RemoteFileSystemView クラス

リモートオブジェクトである `FileSystemView` オブジェクトを取得するクラス

RemoteFileObj クラス

リモートオブジェクトである `File` オブジェクトを取得するクラス

RemoteFileSystemImpl クラス

`FileSystemView` のリモートインターフェース実装クラス。ファイルシステムの情報を返す。

RemoteFileImpl クラス

`File` オブジェクトを返すクラス。ファイルの情報を返す。

アプリケーションからのファイル要求を受けた時の送受信の様子を図 2 に示す。

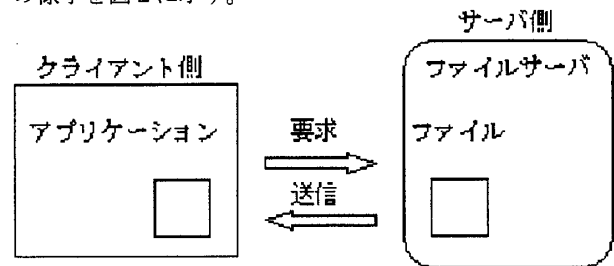


図 2 : ファイルの送受信

5. むすび

本論文ではテキストエディタを用いてリモートファイルシステムの実装について述べた。

現時点でのアプリケーションはテキストエディタのみであるが今後、初心者向けのワードプロセッサや表計算等のアプリケーションを開発・提供する予定である。

◆参考文献◆

[1]: Java RMI 分散ネットワークプログラミング
日本サン・マイクロシステムズ株式会社(編)
発行 株式会社サイエンス社 1998

[2]: Java RMI
ウィリアム グロソ (著), William Grosso (原著),
田和 勝 (翻訳) 発行 オライリー・ジャパン 2001

[3]: 独習 Java
ジョゼフ オニール (著), Joseph O'Neil (原著),
トップスタジオ (翻訳), 武藤 健志 (その他)
発行 翔泳社 1999

[4]: Java™ 2 SDK, Standard Edition ドキュメント V1.4.0
Copyright © 1995-2001 Sun Microsystems, Inc.

† 東洋大学工学部情報工学科