

webMathematica について

高橋 正

神戸大学発達科学部*

1 はじめに

今回の特集は、フリーで使える数式処理システムの紹介である。その意図からすれば、Mathematica は商用のソフトウェアであり、その範疇に入らない。しかし、昨年リリースされた webMathematica は、Java 環境を備えた Web ブラウザーから Mathematica の使用を可能にした。したがって、Mathematica を購入しなくても、その機能を使い、結果を Web ブラウザー上で得ることができる点においては、フリーで使用することができる。したがって、今回の特集に webMathematica の紹介を書かせていただくことにした。

2 Web アプリケーション

Web アプリケーションとは、従来、クライアント側で実行するアプリケーションとは違い、WWW (World Wide Web) を介してサーバにあるアプリケーションを実行し、利用する仕組みである。インターネットが普及する以前から、ネットワークによるそのようなシステムは存在し、金融機関や生産ラインの管理等に使われていた。当初はホストコンピュータと専用端末を接続する形式であったが、やがて安価なパソコンを利用したクライアント/サーバシステムが主流になった。典型的なモデルとしては、ユーザがクライアントマシンから直接ネットワークを介してデータベースに接続、データを要求し、得たデータをクライアントマシンにあるアプリケーションが処理するという方法であった。その後、WWW ブラウザを利用した Web アプリケーションが開発された。そのような機能を提供する事業者を ASP (Application Service Provider) と呼ぶ。今や一般的なものとなったオンラインでのチケット予約やショッピングカート、検索サービスなどもこれに含まれる。Web アプリケーションの最大の特徴は、クライアントにプログラムを配信する必要がないことである。ユーザは、コンピュータにプレインストールされている Web ブラウザとインターネット接続環境さえあ

*takahasi@kobe-u.ac.jp

れば、いつでも Web アプリケーションを利用することができる。また Web アプリケーション提供者 (ASP) は、クライアントのシステムを気にすることなく、Web サーバを立ち上げるだけで世界中のユーザにサービスを提供することができる。従来のシステムに比べ、クライアントプログラムを開発する必要がなくなり、通信手順が HTTP (Hyper Text Transfer Protocol) インターフェースが HTML (必要なら JavaApplet や Macromedia Flash 等でもよい) サーバが Web サーバに統一されたためにユーザの利用環境が容易になった。ただ、このようなシステムはインターネットを介するため、セキュリティ対策は十分に必要がある。数式処理システムにおいても、Web アプリケーション化が求められてきた。代表的な数式処理システムである Mathematica を開発している Wolfram Research Inc. は、平成 13 年 10 月 10 日に webMathematica という Web アプリケーションをリリースした。

3 Tomcat

Tomcat は、Jakarta プロジェクトのサブプロジェクトとして生まれ、現在も開発が続けられているサーバアプリケーションである。Tomcat は Java で記述されており、JSP (Java Server Page) 環境を持った Servlet コンテナと考えることができる。Servlet コンテナはユーザを代表して Servlets を管理したり、呼び出したりする。Servlet コンテナの役割は以下の 3 つに分けることができる。

(1) スタンドアロン型の Servlet コンテナ

JavaWebServer のように、Java ベースの Web サーバであり、単体の Web サーバとして機能する。

(2) プロセス内の Servlet コンテナ

この Servlet コンテナは Web サーバのプラグインと Java インプリメーションの組み合わせである。Web サーバプラグインが JVM 中のアドレス空間をオープンし、Java コンテナはそこで動く。あるリクエストが Servlet で発生した時、リクエストを受け取って、プラグインは (JNI を使って) Java コンテナを制御する。コンテナはマルチスレッド・シングルプロセスで動作するため、多くのリクエストを同時に処理するときなどに効率的な動作が期待できる。

(3) プロセス外 Servlet コンテナ

この Servlet コンテナは Web サーバのプラグインと Web サーバの外部で動いている JVM の Java インプリメンテーションでの組み合わせである。Web サーバプラグインと Java コンテナの JVM は IPC の仕組み (普通は TCP/IP ソケット) で接続する。Servlet に要求があるとプラグインは (IPC を使って) Java コンテナを制御する。外部プロセスエンジンのレスポンスは内部プロセスのものよりも当然良くはないが、安定性においては良い結果が得られる。

Tomcat は Apache や IIS などの他の Web アプリケーションのアドオンとしても組み込むことができ、JSP と呼ばれる HTML 埋め込み型記述スクリプト言語により動的な Web ページを容易に作成することができる ([1, 2, 3])。

4 webMathematica

webMathematica は、数式処理を行う Web アプリケーションである。数式処理システムは、数式処理、数値計算、グラフィックス処理などを統一的に扱えるシステムになった。これまでは、主に科学技術関係の専門家に使われていたのに対し、学校教育の現場でも多くの使用が試みられている。最近では、高校から大学院に至るまで数式処理システムが使われている科目もあり、理系に限らず多くの学生にとって重要なツールとなった。数式処理システム Mathematica は、バージョン 4.1 で MathML や HTML の統合により、インターネットを使った情報交換がより手軽になった。さらに、J/Link を使うことで Mathematica と Java を結合させたアプリケーションの開発が可能になった ([4])。

5 J/Link

Java プログラムから Mathematica Kernel をコントロールすること、及び Mathematica から Java プログラムを実行することが可能になった。この技術は、J/Link という名称である。J/Link テクノロジーは、2000 年 2 月に Wolfram Research によって発表された。それは、Mathematica と Java を統合する MathLink テクノロジーの接続部分である。この技術は、Java クラスとメソッドを Mathematica から用いることができる。そして、Java プログラムからの計算のために、Mathematica カーネルを制御できる。この例として、これまでのものでイメージするには、以前から Wolfram Research の Web サイトで、Web アプリケーション「Integrator」があった。これに似たものを Java で簡単に作成できるようになったと考えるとよいであろう。サーバの Mathematica カーネルをクライアントからの計算要求を計算して、その結果をそのクライアントに返すプログラムである。J/Link は、簡単にこの種類の Web アプリケーションを作ることを可能にする。

このアプリケーションの 2 つの利点がある。

(1) Mathematica がローカルなマシン上にインストールされていないとしても、ユーザはインターネットもしくはローカル・エリア・ネットワークを通して Mathematica の機能を使うことができる。

(2) Mathematica を使う方法を詳しく知らなくても、Mathematica を使った計算結果を知ることができる (Mathematica のコマンドを知らなくても、ボタン操作などで使うことができる)。

現在、Web の使用は、ほとんどの学生にとって簡単なものになった。Web アプリケーションとしての Mathematica の使用は、より簡単に、そして、効果的に数式処理システムを用いることを支援する。

6 webMathematica のメカニズム

サーバ上で Java servlet プログラムを実行するためには servlet エンジンが必要とする。最初に、ユーザは、Web ブラウザの上の変数もしくは式等の、いくつかの必須のパラメータを入力し、Web サーバに送る。サーバはその情報を受ける。そして、servlet エンジンは servlet プログラムを実行する。servlet プログラムは、J/Link を通じて Mathematica カーネルと通信する。カーネルは演算を行い、いくつかの必要なパッケージを呼び出す。その後、クライアントは、計算結果を Web ブラウザで受け取る。

このステップの中で、servlet は、ユーザの Web ブラウザの上で、その結果を表示するためにダイナミックな HTML ファイルを生成する。servlet を単純にして、カーネルへのアクセスの数を減少させるために、全部のプロセスを、そのカーネルの中で扱うパッケージを走らせること等が、この技術の効果的な使用方法である。これまでの使用において指摘されている、webMathematica の問題点として、以下のことが挙げられる。

速度：データサイズが大きくなるため、アニメーションをするためには、長い時間がかかる。データを圧縮するか、コード化することが重要である。

安全：ファイルシステムを取り扱うので、Mathematica は危険な機能を持ってしまう。J/Link は任意の Java クラスとメソッドを使うことを許す。そのため、セキュリティシステムをセットアップすることが重要である。

メンテナンス：servlet プログラムを書くために Java プログラミングについての知識を必要とする。また、それを書き直すたびに、それを再コンパイルしなければならない。

この技術によって、servlets を詳しく知らなくても、Web アプリケーションをつくることができる。このシステムは、セキュリティシステムを持っている。J/LINK の問題点で述べた安全とメンテナンスについて、Link アプリケーションは、MSP システムを使うことによって保たれる。webMathematica は、Java Servlets の技術に基づくものである。

webMathematica は、MSP (Mathematica Server Page) と呼ばれる独自の HTML ソース埋め込み型言語によりビジネスロジック処理を行う。まず、クライアントのブラウザが、Web サーバ (Tomcat) 上にある MSP ソーススクリプトを URL で指定し、リクエストする。Web サーバはリクエストされたファイルの拡張子が msp であると識別すると、MSP エンジンに処理を渡す。MSP エンジンはリクエストされた MSP ソーススクリプトを、リクエスト時にヘッダと共に送られてきたパラメータを元にコンパイルし、Java サブレットを生成する。そして、サブレットが実行され、サブレットは J/Link クラスを使いサーバの Mathematica カーネルに演算処理をさせ、実行結果を受け取る。その実行結果はサブレットから Web サーバ (Tomcat) に渡され、Web サーバ (Tomcat) は HTML (MathML や XML) 形式にしてブラウザに返し、プロセスを終了する (図 1 を参照)。

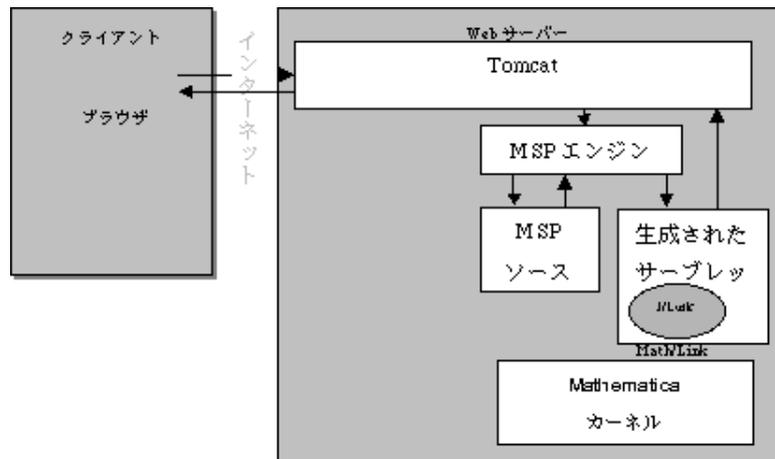


図 1: webMathematica のメカニズム

単純な MSP のソース例を以下に示す (variable.msp ソース)。

```
<HTML>
<HEAD>
<TITLE>Assigning Variables</TITLE>
</HEAD>
<BODY BGCOLOR="#ffffff" >
<H1>Assigning Variables</H1>
<FORM ACTION="Variables" METHOD="POST">
Enter something:
<INPUT TYPE="TEXT" NAME="input" ALIGN="LEFT" SIZE="10" >
<br>
<%Mathlet $$input %>
<br>
<INPUT TYPE="Submit" NAME="submitButton" VALUE="Evaluate">
</FORM>
</BODY>
</HTML>
```

この variable.msp ソースを Web サーバ上の適切な場所に置き、ブラウザから URL を指定する。このソースをブラウザによりリクエストした結果表示される画面は次のようなものである (図 2 を参照)。

このページ自体は、HTML 言語で書かれており、フォームと呼ばれる入力を受け付ける画面が表示される。これに、以下のように数値を入力し、Evaluate ボタンを押す。この HTML

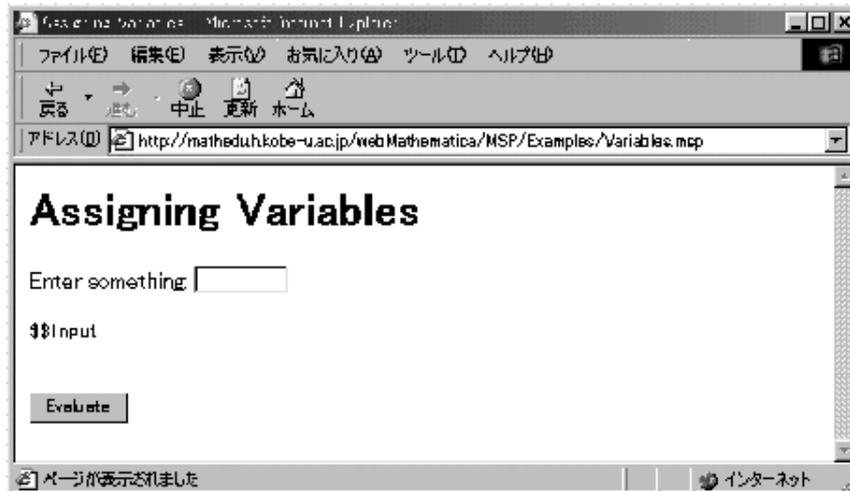


図 2: webMathematica のブラウザ画面

ページのフォームタグはこのスクリプトを指定しており、Evaluate ボタンを押すことによっていくつかの隠しパラメータとフォームに記入された値がヘッダと共にサーバに送られる(図 3 を参照)。

すると、MSP プログラム内の input 変数に値が入り、それを表示させる(図 4 を参照)。

7 webMathematica のインストール

本稿では、LinuxOS 上における webMathematica のインストール作業について説明する。

7.1 JDK のインストール

JDK のインストールは root 権限で行う。jdk-1.3.1.i386.rpm でインストールを行う。
/usr/java/jdk1.3.1/ に展開される。

7.2 J/LINK のインストール

J/LINK のインストールは root 権限で行う。JLink v112 linux.tar.Z を展開した JLink を

/usr/local/mathematica/AddOns/Applications/ に移動する。

/usr/local/mathematica/AddOns/Applications/JLink/SystemAdditions/

の JLink.jar を /jdk へのパス /jre/lib/ext/(/usr/java/jdk1.3.1/jre/lib/ext/) に移動する。



図 3: webMathematica の入力

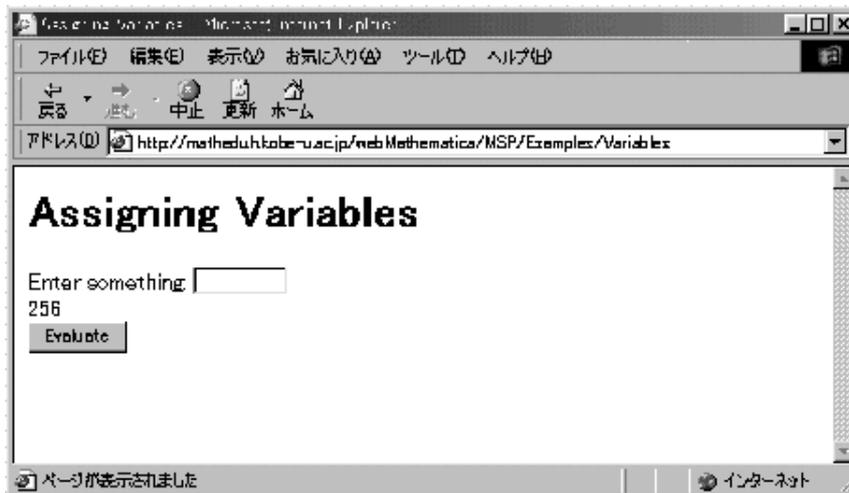


図 4: webMathematica の出力

```
chown root.bin /usr/java/jdk1.3.1/jre/lib/ext/JLink.jar

```

を実行して所有グループを変更する。Java 1.2 以降なので、
 /usr/local/mathematica/AddOns/Applications/JLink/SystemAdditions/linux/
 の libJLinkNativeLibrary.so を /jdk へのパス
 /jre/lib/<arch>(/usr/java/jdk1.3.1/jre/lib/i386/) にコピーする。
 chown root.bin /usr/java/jdk1.3.1/jre/lib/i386/libJLinkNativeLibrary.so
 chmod 755 /usr/java/jdk1.3.1/jre/lib/i386/libJLinkNativeLibrary.so
 を実行して所有グループと属性を変更する。

7.3 Tomcat のインストール

Tomcat のインストールは root 権限で行う。jakarta-tomcat-3.2.3.tar.gz を展開し、
 jakarta-tomcat-3.2.3 を tomcat とリネームする (以下の内容は tomcat へのリネーム
 を前提とする)。tomcat/bin/ の tomcat.sh, startup.sh, shutdown.sh を chmod 755
 xxx.sh で実行可能にする。

7.4 MSP のインストール

MSP のインストールは root 権限で行う。MSP-0.93.tar.gz を展開した MSP を
 /usr/local/mathematica/AddOns/Applications/ に移動する。
 webMathematica-0.93.tar.gz を展開した webMathematica を tomcat/webapps/ に移動
 する。

/usr/local/mathematica/AddOns/Applications/MSP/Configuration/MSP.conf の
 KernelAcquireLimit=2000 を KernelAcquireLimit=200 に、
 VerboseLogs=false を VerboseLogs=true に、それぞれ変更し、
 MSPIncludesDirectory=/webMathematica/MSPIncludes
 を追加する。

tomcat/conf/server.xml に

```
<Context path="/webMathematica"
    docBase="webapps/webMathematica"
    crossContext="false"
    debug="0"
    reloadable="true" >
</Context>
```

を追加する。

ディレクトリ /usr/local/mathematica/AddOns/Applications/MSP/Caches
 の属性を tomcat が書き込み出来るように変更する。

/usr/local/mathematica/AddOns/Applications/MSP/Configuration

/Licensing/mathpass

を編集する (/usr/local/mathematica/Configuration/Licensing/mathpass の内容のコピーで良い)。

参 考 文 献

- [1] Jakarta プロジェクト, 「Tomcat3.2.3 日本語ドキュメント」(2001),
<http://www.ingrid.org/jajakarta/site/tomcat/tomcat-3.2.3/tomcat-3.2.3-doc-ja.zip>.
- [2] Marty Hall (岩谷宏 訳), コア・サーブレット & JSP -Java サーバ技術による Web 開発-, SOFTBANK Publishing, 2001.
- [3] ジェームス・グッドウィル (今野睦 訳), JSP クイックリファレンス -Java Server Page の世界-, ピアソン・エデュケーション, 2001.
- [4] <http://www.wolfram.com/products/webmathematica/>