

webMathematica とデータベースの連携技術について

若松 隆*

日本テレコム株式会社 サービス開発本部†

1 はじめに

急速なブロードバンドインターネットの普及によって、常時接続・高速インターネット環境は、より身近なものとなった。それに伴い、従来クライアントマシン側でおこなってきた演算処理を、インターネットを介してサーバ側で行う「Web アプリケーション」という形態が注目されはじめ、今ではインターネット上の多くのサービスは Web アプリケーションで行われている。ここ数年、パーソナルユースのコンピュータには、付加価値をつけるために、たくさんのアプリケーションがプリインストールされるようになった。アプリケーションの多様化に伴い、ユーザの使いたいアプリケーションを Web アプリケーションで提供することには意義がある。そして Web アプリケーションは、サーバ側アプリケーションと連携することによってさらに強力なツールとなる可能性を持っている。その中でも、特に、データベースサーバとの連携が挙げられる。Web アプリケーションはデータベースサーバと連携することにより、インターネットの特性を生かし、集められた多くのデータを蓄積・再利用することができる。

数式処理システムの世界でも、このような Web アプリケーション化の流れが出てきている。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。本研究では、Web アプリケーションである webMathematica の特徴を分析し、応用利用としてより強力なサービスを利用者に提供するための「データベースサーバとの連携手法」について考察し、数学教育に効果的なテストシステムの作成を行うことを目的とする。

*takashi.wakamatsu@japan-telecom.co.jp

†神戸大学大学院総合人間科学研究科前期博士課程 (平成 14 年 3 月修了)

2 準備

2.1 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 ページを容易に作る ([1, 2, 3])。

2.2 RDBMS

RDBMS (リレーショナルデータベース管理システム) とは、リレーショナルデータベースを管理するためのソフトウェアである。リレーショナルデータベースとは、一件の項目を複数の項目 (フィールド) として認識し、そのデータ集まりを表の概念でテーブルという単位でデータの保存、蓄積を行う。RDBMS は、ID 番号やキーとなるフィールドのデータを利用して、高速にデータの統合や抽出を行うことができる。データベースとしてはこの方式が最も広く使われており、大規模システムでは Oracle 社の「Oracle」、小規模システムではマイクロソフト社の「Access」が広いシェアを誇る。近年ではフリーながらも高い信頼性を誇

る PostgreSQL や MySQL などの人気も高まっている ([4])。

2.3 PostgreSQL

PostgreSQL は、フリーなオブジェクト指向 RDBMS であり、主要な UNIX 上で動作する。名前からわかるようにデータベース標準言語 SQL92 に対応しており、さらに独自の強力な機能により近年人気が出てきている。

2.4 PHP

PHP (正式名称は Hypertext Preprocessor) は、HTML 埋め込み型のサーバサイド・スクリプト言語である。C 言語ベースで書かれており、Apache Web Server にモジュールとして組み込んで使う方法が一般的である。PHP の特徴には以下のようなものが挙げられる。

- ・ スクリプト言語

PHP はコンパイルを必要とせず、ソースを修正してすぐにテストすることができる生産性の高いスクリプト言語である。その反面、実行のたびに文法解析が行われ、実行速度が落ちてしまうというリスクを負うが、Apache モジュールとして動作することによってそのリスクは軽減される。

- ・ HTML への埋め込み型言語

PHP はファイル全体をスクリプトとして作成する必要はなく、HTML ページのなかのスクリプト処理したい部分だけを PHP で書くという利用法をとることができる。HTML で書かれた構文のなかに `<? ? >` で囲まれたスクリプトを書くことにより、実行時にその部分を必要な出力などと変換して HTML 文書をクライアントに出力する。

- ・ C ライクな文法

if, for, while など、C 言語でよく使われるものと同じ制御構文が用意されており、ユーザ定義関数、ほかのライブラリをインクルードして使うことも可能である。

- ・ Perl 互換の強力な正規表現

Perl の特徴である強力な正規表現機能を備えており、リスト配列、連装配列、多次元配列もサポートされている。

- ・ Apache のモジュールとして動作

CGI (Common Gateway Interface) とは違い、Apache の一部として動作させることができるため、無駄なリソースを消費することなく、処理も高速である。

- ・ 各種データベースへのインターフェース

PHP は Oracle, Sybase, Informix などの商用 DBMS と、PostgreSQL, MySQL 等のようなフリーの DBMS へのインターフェースを標準で備えており、容易に接続することができる ([5, 6])。

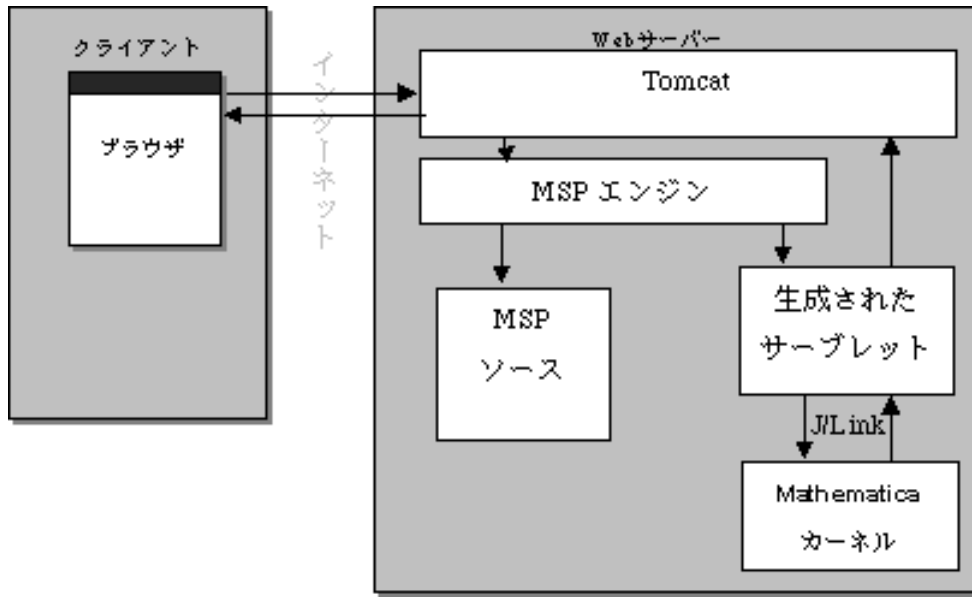


図 1: webMathematica のメカニズム

2.5 webMathematica のメカニズム

webMathematica は、数式処理を行う Web アプリケーションである。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。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)。

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 ([7])。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てプロセスを終了する (図 1 参照)。

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ぶ。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。

3 データベースとの連携手法

「webMathematica とデータベースが連携する」とは具体的には、データベースに保存した内容を任意に取り出し、プログラムによりその値を利用して webMathematica に演算させるということである。この条件を満たす手法として、以下に 4 つのケースを提案する。その後、それぞれのケースについて検討し、webMathematica の特徴を有効に使い、効率的にデータベースと連携できる手法を考察する。

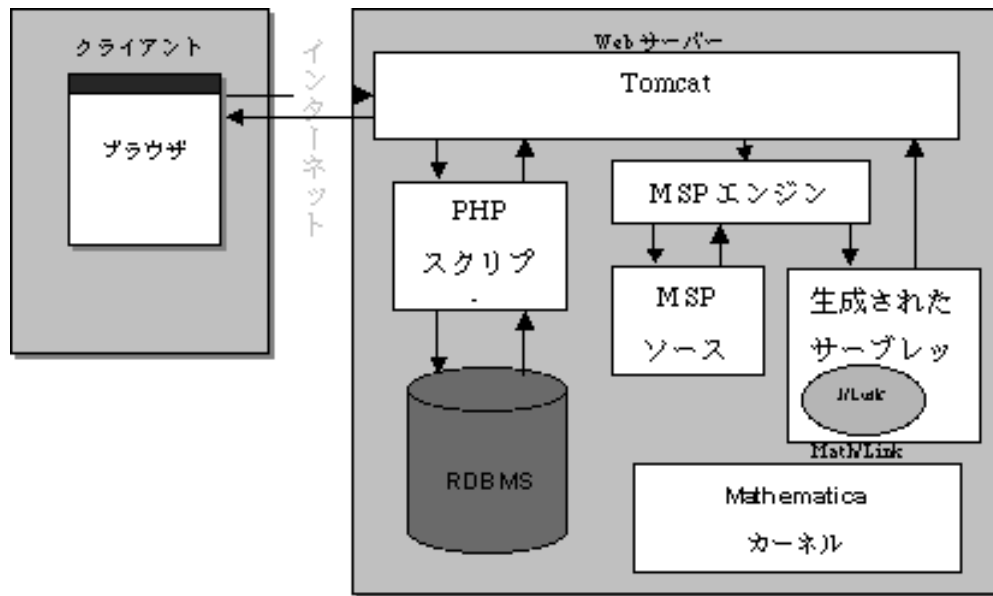


図 2: 連携手法 1

3.1 連携手法 1

この手法は PHP スクリプトを利用する。

この手法は、まずブラウザからのアクセスを受けた PHP スクリプトが、データベースエンジンにアクセスし、情報を整理して HTML をクライアントに返す。その際、HTML には隠しパラメータとしてデータベースから得た値を埋め込んでおく。その後クライアントからの再アクションを行うと、ブラウザはデータベースの内容をヘッダと共に保持したまま webMathematica にリクエストを行う。その後、その値を利用し webMathematica がクライアントに処理結果を渡す。それによりクライアントの処理結果に、データベースの内容を反映することができる (図 2 参照)。

3.2 連携手法 2

この手法は、連携手法 1 と同じく PHP を利用する。

ブラウザからのアクセスを受けた PHP スクリプトはデータベースにアクセスし、その値を受け取る。次に PHP スクリプトはその値を利用し、あらかじめ決められた書式で MSP スクリプトを生成する。そして同時にクライアントには、リダイレクトヘッダを転送する。リダイレクトヘッダを受け取ったクライアントは、自動的に、生成された MSP ファイルを指定してアクセスする。生成された MSP ファイルには既にデータベースからの値を保持しており、その値を利用して適切な演算処理をするように記述されているため、クライアントにデータベースの値が反映された処理結果を返すことができる (図 3 参照)。

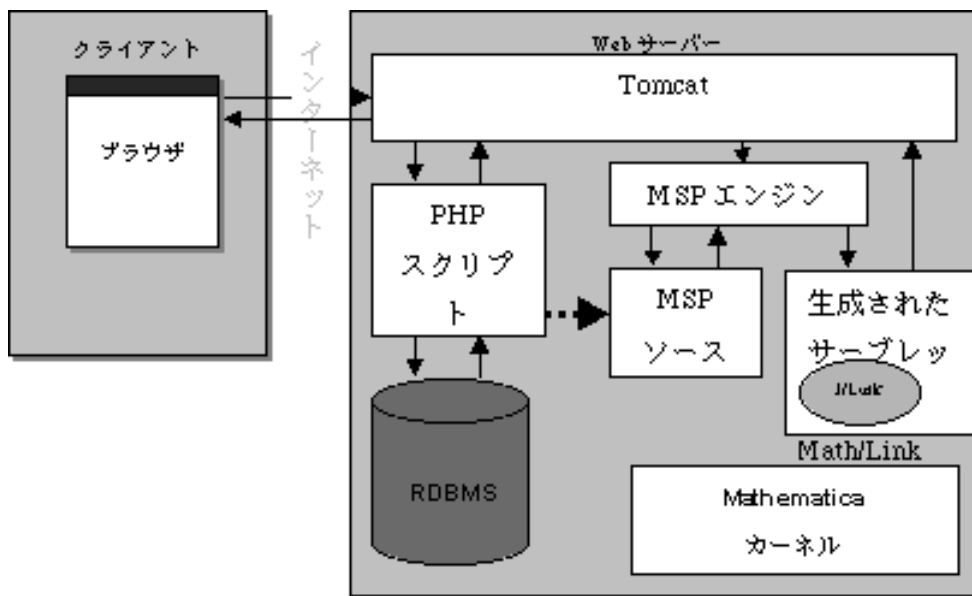


図 3: 連携手法 2

3.3 連携手法 3

この手法では、Mathematica カーネルに Database Access Kit をインストールしたものを利用する。

Database Access Kit は、Wolfram Research Inc. からリリースされている、Mathematica にデータベースへアクセスする機能を拡張するソフトウェアである。この機能拡張により、Mathematica コマンドによって、Mathematica が直接データベースにアクセスする。そして Mathematica はデータベースから抽出した値をもとに演算し、その結果を MSP エンジンに渡す。そして MSP エンジンは HTML を生成し、クライアントに計算結果として返す。Mathematica が直接 DB と通信するため、MSP ソースに Database Access Kit 用の拡張関数を利用することによってそのまま DB の内容を利用し演算結果を返すことができる（図 4 参照）。

3.4 連携手法 4

この手法は、Java により MSP の機能を拡張する。

MSP エンジンはそもそも Mathematica と Tomcat のあいだをとりもつための Java Servlet で書かれたプログラムである。MSP は、J/Link ライブラリにより拡張されており、Math/Link というプロトコルで Mathematica と通信することができる。また、処理結果を Tomcat に渡すことにより、HTML を生成してクライアントに返す。前述のように MSP は JavaServlet によって作られたプログラムであり、ライブラリをインクルードすることによって機能を拡

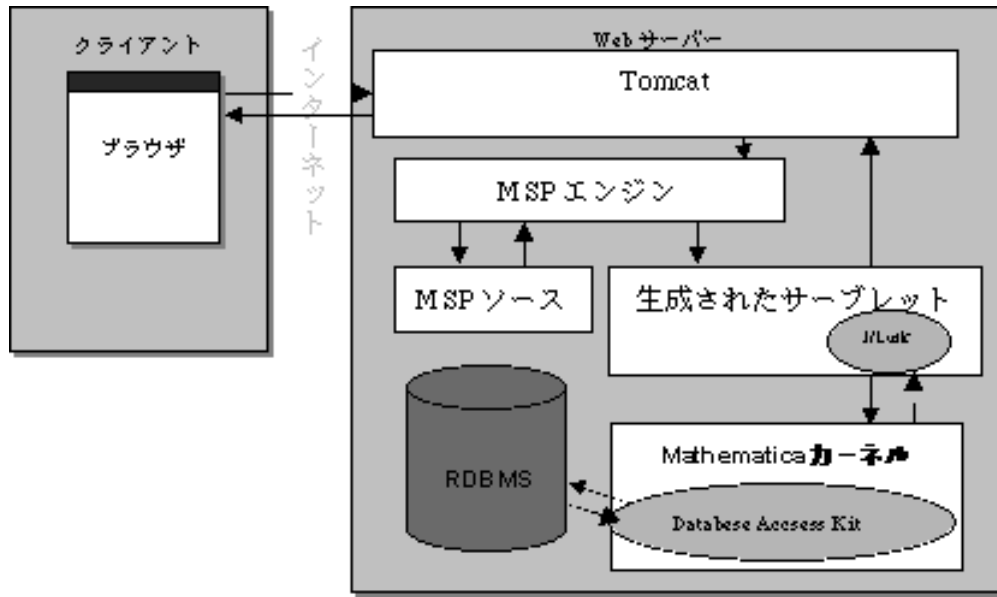


図 4: 連携手法 3

張できる。すでに MSP は J/Link というライブラリを標準で取り込む仕様になっているが、同様に新しくデータベースと通信するライブラリ (JDBC) を取り込むようカスタマイズすることができれば、一度のリクエストでデータベースと Mathematica と通信することができる。ただ、データベースと Mathematica カーネルそれぞれにアクセスする時間のあいだ、プログラムはプロセスを一つ占有するため、サーバへの負荷が大きい (図 5 参照)。

3.5 連携手法に関する考察

ここまで挙げた 4 つの手法についてそれぞれの特徴とメリット、デメリットをユーザと開発者の面から考察する。まず手法 3 の Database Access Kit を Mathematica に組み込む手法については、Database Access Kit は Windows 系の IIS サーバ版しかリリースされておらず、今回構築する UNIX-OS では動作しない。また、Database Access Kit は、Wolfram Research Inc. から発売されている高価なシェアウェアである。

手法 4 は Java による MSP エンジンの機能拡張である。一般的に小規模なシステムでは、同じサーバ上に Web サーバとデータベースサーバが同居していることが多い。現状ではまだ webMathematica は同時に大量の演算処理リクエストをこなすことができない。これはサーバ内での Mathematica カーネルプロセス数の制限などが影響していると思われる。手法 4 は一度のプロセスでまずデータベースにアクセスし、次に Mathematica カーネルにアクセスする。そのため各アプリケーションサーバからの返答を待っている間、プロセスが占有されてしまうため、パフォーマンスが落ちてしまう。手法 1, 2 は、どちらも 2 つのフェーズ

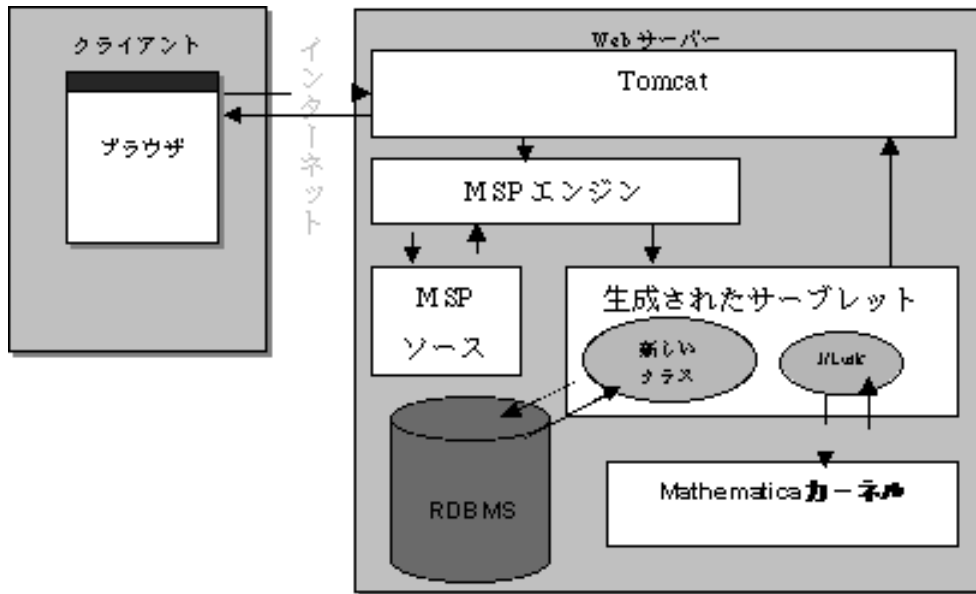


図 5: 連携手法 4

に分かれている。PHP 言語によりデータベースにアクセスし、データを取り出すフェーズと、そのデータをもとに Mathematica に演算処理をさせ、クライアントに結果を表示するフェーズである。数式処理演算 Web アプリケーションとデータベースを連携する際、構築したいコンテンツの種類により様々な柔軟性が求められる。特にデータベースを利用した Web アプリケーションは、一度取り出した値をクライアントに返してから選択させ（一覧表示などで）、そしてその値をキーにしてさらに他のカラムの値を利用するケースが多い。もちろんクライアントからの一度のアクションで、データベースの値を利用するケース（現在の DB 内容を用いてのグラフ表示等）もある。

Web アプリケーションはユーザの利用方法を限定することにより、少ないアクションで多くの機能が利用できる。その反面、多種多様なコンテンツ作成の技術が、コンテンツ作成者に求められる。しかしながらコンテンツ作成における人的コストを節約するため、できるだけ開発効率の高い（作成者がすぐに覚えることができる）手法を選択しなければならない。大規模なデータベースを利用したシステムでは、データベースの値を一度取り出してから次のアクションに移るケースが圧倒的に多く、ユーザにもわかりやすい。よって毎回 MSP スクリプトを生成する手法 2 よりも、手法 1 のほうが応用が利き、かつ PHP と MSP のスクリプト開発が別れているため、現状では、開発効率がよいと考えられる。

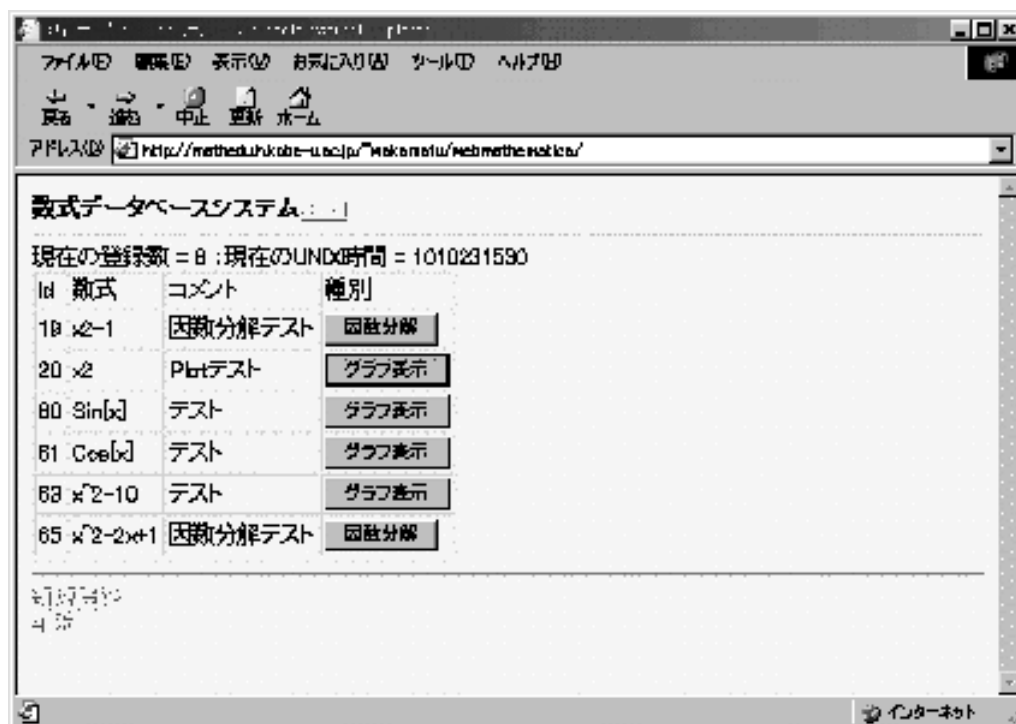


図 6: デフォルトページ

4 テストシステムの作成

本章では、前章における連携手法 1 を利用したテストシステムの作成を行なう。テストシステムは連携手法 1 の、「一度データベース内の情報を取りだし内容をクライアントに提示してからユーザに選択させる」という特徴を生かすため、『データベースに数式とパラメータを登録しておき、必要な項目を一覧・検索表示し、選択した数式のグラフ化や因数分解など登録者の意図したアクションを起こすシステム』を作成する。

4.1 テストシステムの概要

この手法において実際に作成しなければならない部分は、データベースから値を取り出すための PHP スクリプトと、実際に Mathematica に計算をさせる MSP スクリプトの 2 つである。この手法のテストシステムとして、「登録された数式を一覧表示し、登録者の意図する処理結果を表示させる」ものを作成する。

作成の流れは以下ようになる。

- (1) データベーステーブルを作成
- (2) 登録データ(数式)を一覧表示する PHP スクリプトの作成
- (3) データ内容を新規登録する PHP スクリプトの作成

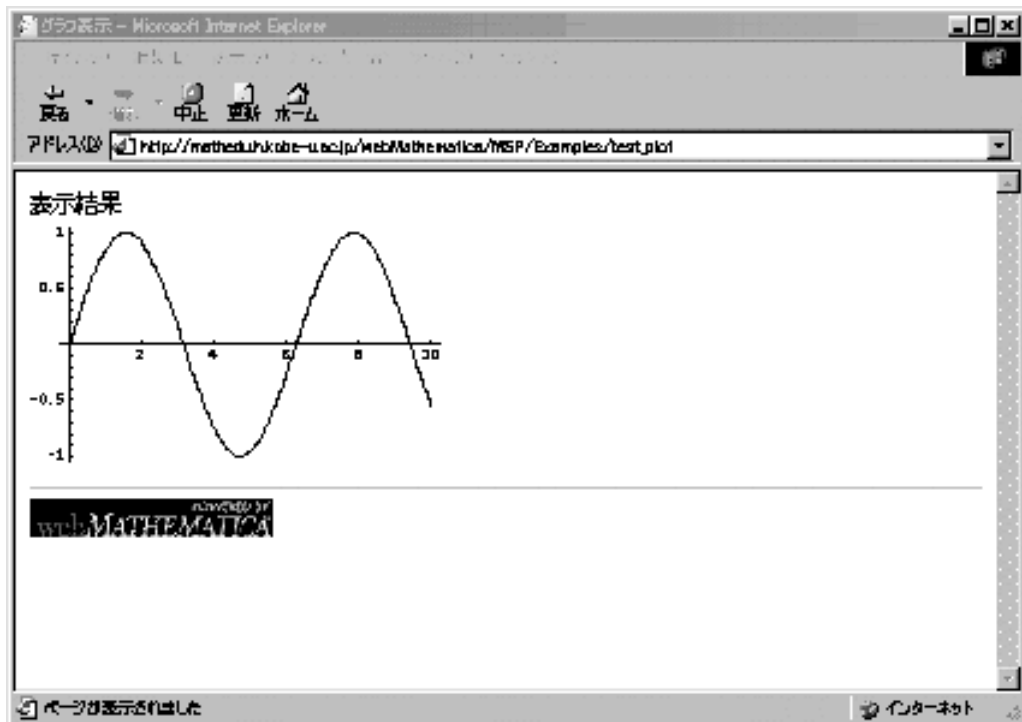


図 7: グラフ描画スクリプト実行後

- (4) データ内容を更新・削除する PHP スクリプトの作成
- (5) 数式を決められた方式で処理し、結果を表示する MSP スクリプトの作成

4.2 Postgres テーブルの作成

データベーステーブルを作成する前に、その設定値を記した以下のテキストファイル math.sql を、サーバ上にアップロードしておく（付録参照）。

PostgreSQL に新しいテーブルを作るには、コマンドライン上で psql と入力し、psql モードにならなければならない。postgres ユーザになり、先ほどのファイル（math.sql）をアップロードしたフォルダに移動してから psql と入力すると、psql モードに入る（図 6，7，8，9 参照）。

4.3 テストシステムの動作

作成した PHP 及び MSP スクリプトをサーバ上に正しく設置し、ブラウザで指定してアクセスすると以下のような表示になる（図 6 参照）。

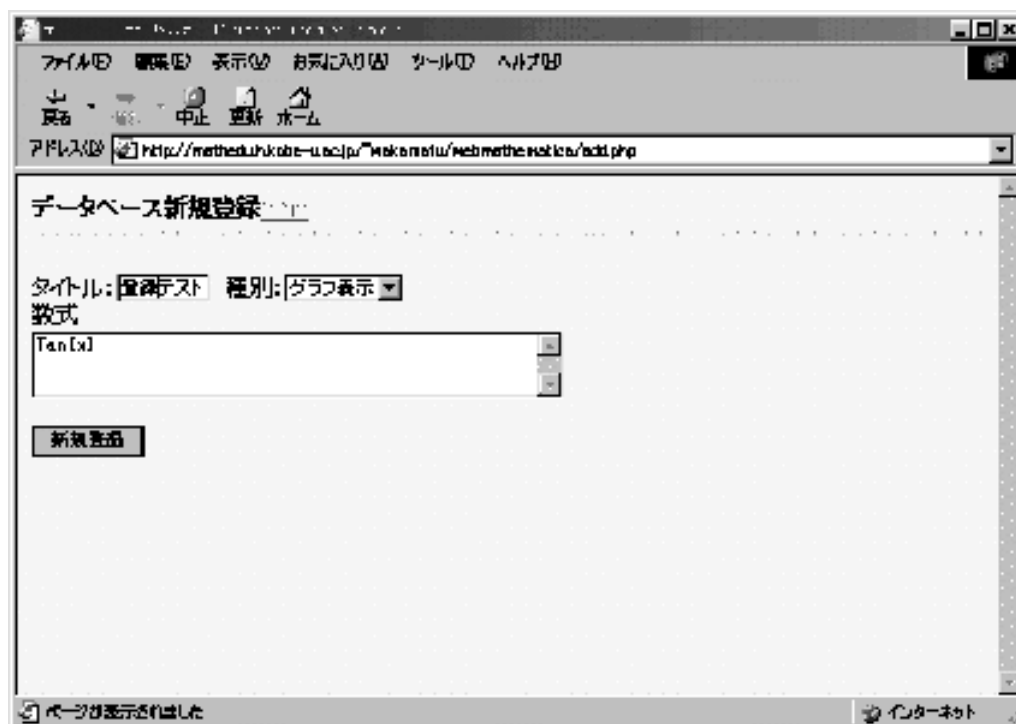


図 8: 新規登録画面

この画面は、PHP スクリプトにより表示されている。この PHP スクリプトがブラウザによりリクエストされると、PHP スクリプトは PostgreSQL データベースにアクセスし、得た情報の一部を HTML に整形し表示する。この画面では、既に登録されている 6 つのデータが id 順にソートされ、一覧表示されている。

次に図 6 に示した各コメントに従い、ブラウザにより操作を行った結果を示す。

図 7 は、数式グラフを描画する MSP スクリプトである。デフォルトページの 3 のボタンを押すと、 $\text{Sin}[x]$ を隠しパラメータで描画 MSP スクリプトに送り、結果を表示する。このグラフは webMathematica カーネルが GIF 画像で出力したものであり、保存しておけば再利用することもできる。

4.4 新規登録画面

画面は PHP スクリプトで描画されており、このフォームに必要な内容を入力して「新規登録」ボタンを押すことにより、データベース登録スクリプトが作動する。以下の図 8 では $\text{Tan}[x]$ をグラフ表示モードで登録している。

図 8 の「新規登録」ボタンを押すとブラウザは自動的にデフォルトページに切り替わり、図 9 のように最新の情報に更新される。id は PostgreSQL の「シークエンス」という機能に

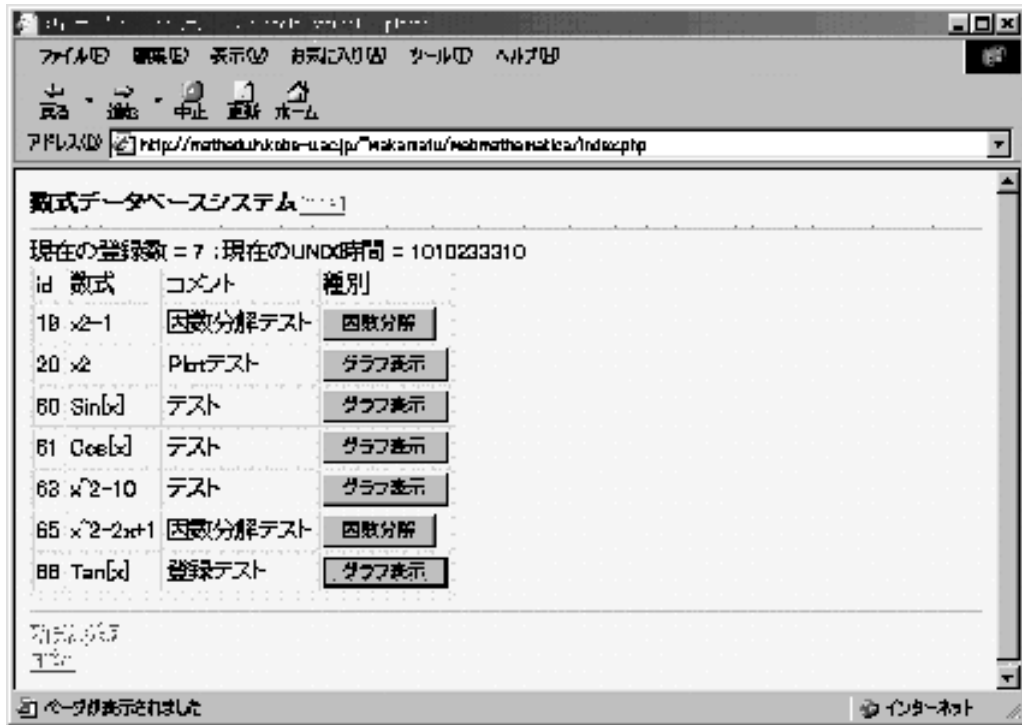


図 9: 新規登録後のデフォルトページ

より別テーブルで保管されており、最新の登録では常に + 1 された数をデータベースの値とする。これは、最新でないデータを削除しても ID がずれないということであり、データベースを使う上で特定のデータがどこにあるか指定する際に便利なキーとなる。

新規登録と同様にデータベースの削除も可能である。id 項目の横にあるラジオボタンにチェックを入れ、「チェックした項目を削除」ボタンを押すことにより PHP スクリプトが動作し、該当する項目を削除することができる。

5 考察及び今後の課題

本研究のテストシステム作成にあたり、webMathematica は正式版ではなく、ベータ版であるバージョン 0.93 を使用した。正式なリリースはすでにされているが、ライセンスなどの関係上、今回は採用することができなかった。本研究では、4 つの連携手法のうち、最もデータベースとの連携を明示的に行うことの出来る連携手法 1 を選択した。しかし、当然作るとうとするアプリケーションの目的や利用方法により、最適な手法は変わってくる。また今後 webMathematica のバージョンが進むことにより、MSP エンジン自体に各種データベース用インターフェースモジュールの組み込まれることが予測される。その場合、連携手法 4 のケースに近いシステムになる。しかし本研究で作成したテストシステムの多くの部分は、今後

も様々な Web アプリケーション開発においてその技術を用いることができる。PostgreSQL データベーステーブルを作る際に用いたテンプレートや PHP 及び MSP スクリプトはどれも、簡単な修正を行うだけで再利用が可能である。webMathematica のように高機能な Web アプリケーションプラットフォームが出現し、より高度な要求をこなせるようになった今、開発効率を高め、汎用性のあるシステムを作成することは重要である。今後は、実用的で複雑なシステムの作成、及び、4つの手法の再検討などを行い、開発者にとって作りやすく、利用者にとって分かりやすいシステムの作成に取り組みたい。

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] 石井達夫, 改訂版 PC UNIX ユーザのための PostgreSQL 完全攻略ガイド, 技術評論社, 1999.
- [5] 堀田倫英, 石井達夫, 廣川類, PHP 徹底攻略-Web とデータベースの連携プログラミング-, SOFTBANK Publishing, 2000.
- [6] 宮元和明, ホンキで学ぼう! PHP のキホン, 秀和システム, 2001.
- [7] <http://www.wolfram.com/products/webmathematica/>

A 付録

```

index.php
<?
//初期設定=====
$dbname = "postgres";//DB名
$dbtable = "math";//テーブル名
$title = "数式データベースシステム";
$adminpass = "root";//管理モード用パスワード
$unixtime = time();//UNIX 時間
$ip = getenv("REMOTE_ADDR");//IP アドレス取得
//セレクト部分 HTML
$select = <<< EOF
<select name="kind">
<option value="1" selected>グラフ表示
<option value="2">因数分解
</select><br>
EOF;
$con = pg_connect("", "", $dbname);
if($con == false){ die("データベース接続に失敗しました。");}

```

```

$sql = "select * from $dbtable order by id";
$result = pg_exec($con,$sql);
$rows = pg_numrows($result);          // 行数を取得
$columns = pg_numfields($result);     // 列数を取得
print <<< EOF
<HTML>
<HEAD>
<TITLE>$title</TITLE>
<META HTTP-EQUIV="Content-type" CONTENT="text/html; charset=euc-jp">
</HEAD>
<BODY bgcolor=fff5ff>
<b>$title</b><font size=2><a href=index.php>[TOP]</a></font><br><hr>
現在の登録数 = $rows : 現在の UNIX 時間 = $unixtime<br>
<Table Border=1 CellSpacing=1 CellPadding=2 BorderColor=#ffbfff>
EOF;
for ($j = 0;$j < $rows;$j++) {
//見出し行
    if ($j == 0) {
        print("<tr><td>id</td><td>数式</td><td>コメント</td>
<td>種別</td></tr>\n");
    }
//データ行
    $arr = pg_fetch_array($result,$j);
//Plot の場合のエラー防止
    if($arr[4] == "1"){
        $arr[4] = "グラフ表示";
        if( $arr[5] == "" ){ $arr[5] = 10; }
    }
    print <<< EOF
    <tr><td>$arr[0]</td><td>$arr[3]</td><td>$arr[1]</td><FORM ACTION="http://
mathedu.h.kobe-u.ac.jp/webMathematica/MSP/Examples/test_plot" METHOD="POST"><td>
<input type="hidden" name="fun" value="$arr[3]">
<input type="hidden" name="x1" value="$arr[5]">
<INPUT TYPE="Submit" VALUE="$arr[4]">
</td></FORM></tr>
    EOF;
}elseif($arr[4] == "2"){
    $arr[4] = "因数分解";
    print <<< EOF
    <tr><td>$arr[0]</td><td>$arr[3]</td><td>$arr[1]</td><FORM ACTION="http://
mathedu.h.kobe-u.ac.jp/webMathematica/MSP/Examples/test_factor" METHOD="POST"><td>
<input type="hidden" name="fun"
value="$arr[3]"><INPUT TYPE="Submit" VALUE="$arr[4]"></td></FORM></tr>
    EOF;
}
}
pg_freeresult($result); // 検索結果の解放
pg_close($con);        // データベースとの接続切断
print <<< EOF
<td></td></table>
<hr>
<a href="./add.php">新規登録</a><br>
<a href="./del.php">削除</a>
</Body></Html>
EOF;
?>
    add.php
<?
//初期設定=====
$dbname = "postgres";          //DB名
$dbtable = "math";            //テーブル名
$title = "データベース新規登録";

```

```

$script = "add.php"; //このスクリプト名
$unixtime = time(); //UNIX 時間
$ip = getenv("REMOTE_ADDR"); //IP アドレス取得
//セレクト部分 HTML
$select = <<< EOF
<select name="kind">
<option value="1" selected>グラフ表示
<option value="2">因数分解
</select><br>
EOF;
if ($mode == add){
$con = pg_connect("", "", "$dbname");
if($con == false){ die("データベース接続に失敗しました。");}
$sql = "INSERT INTO $dbtable(head,body,mathml,kind,value1,value2,value3,
value4,value5,value6)
VALUES('$head','$body','$mathml','$kind','$value1','$value2','$value3',
'$value4','$value5','$value6')";
$result = pg_exec($con,$sql);
pg_freeresult($result); // 検索結果の解放
pg_close($con); // データベースとの接続切断
header("Location: ./index.php");//処理が終了したらトップへ
//データベースから削除
}
else{
print <<< EOF
<HTML>
<HEAD>
<TITLE>$title</TITLE>
<META HTTP-EQUIV="Content-type" CONTENT="text/html; charset=euc-jp">
</HEAD>
<BODY bgcolor=fff5ff>
<b>$title</b><font size=2><a href=index.php>[TOP]</a></font>
<hr>
<form method=POST action="$script" target="_top">
<input type="hidden" name="mode" value="add">
タイトル:<input type="text" name="head" size=10 value="テスト"> 種別: $select
数式<br>
<textarea name=mathml rows=3 cols=50>
MathML で入力可能
</textarea><br>
<br>
<input type=submit value="新規登録">
</form></Body></Html>
EOF;
}
?>
del.php
<?
//初期設定=====
$dbname = "postgres"; //D B名
$dbtable = "math"; //テーブル名
$title = "データベース新規登録";
$script = "add.php"; //このスクリプト名
$unixtime = time(); //UNIX 時間
$ip = getenv("REMOTE_ADDR"); //IP アドレス取得
if($mode == del){
$con = pg_connect("", "", "$dbname");if($con == false){ die("データベース接続に失敗しました。");}
$sql = "DELETE FROM $dbtable WHERE id = $id";
$result = pg_exec($con,$sql);
pg_freeresult($result); // 検索結果の解放
pg_close($con); // データベースとの接続切断
}

```



```

header("Location: ./index.php");//処理が終了したらトップへ
}else{
$koumoku = array('項目 ID','項目名','本体','MathML','種別','値1','値2','値3',
                '値4','値5','値6');
$con = pg_connect("","$dbname");
if($con == false){ die("データベース接続に失敗しました。");}
$sql = "select * from $dbtable order by id";
$result = pg_exec($con,$sql);
$rows = pg_numrows($result); // 行数を取得
$columns = pg_numfields($result); // 列数を取得
print <<< EOF
<HTML>
<HEAD>
<TITLE>$title</TITLE>
<META HTTP-EQUIV="Content-type" CONTENT="text/html; charset=euc-jp">
</HEAD>
<BODY bgcolor=fff5ff>
<b>$title</b><font size=2><a href=index.php>[TOP]</a></font>
<hr>
現在の登録数 = $rows : 現在の UNIX 時間 = $unixtime<br>
<Table Border=1 CellSpacing=1 CellPadding=2 BorderColor=#ffbfff>
<form method=POST action="./del.php" target="_top">
<input type="hidden" name="mode" value="del">
EOF;
for ($j = 0;$j < $rows;$j++) {
//見出し行
if ($j == 0) {
print("<tr>");
for ($i = 0;$i < $columns;$i++) {
$str = pg_fieldname($result,$i); // 列名の取り出し
print("<td><font size=2><b>$koumoku[$i]</b><br>($str)</font></td>");
}
print("</tr>\n");
}
//データ行
print("<tr>");
for ($i = 0;$i < $columns;$i++) {
if ($i == 0){
$str = pg_result($result,$j,$i);// データの取り出し
print "<td><input type=radio name=id value=$str>$str</td>";
}elseif($i == 3){
$str = pg_result($result,$j,$i);// データの取り出し
$str = htmlspecialchars($str);
print "<td>$str</td>";
}else{
$str = pg_result($result,$j,$i);// データの取り出し
print("<td>$str</td>");
}
}
print("</tr>\n");
}
pg_freeresult($result); // 検索結果の解放
pg_close($con); // データベースとの接続切断
print <<< EOF
</table>
<input type=submit value="チェックした項目を削除">
</form></Body></Html>
EOF;
}
?>
test_factor.msp

```

```

<HTML>
<HEAD>
<TITLE>因数分解</TITLE>
</HEAD>
<BODY text="#171717" BGCOLOR = "#ffffff">
因数分解実行後<br>
<table border="0" cellspacing="0" cellpadding="2" bgcolor="#c0c0c0">
<tr>
<td align="center" valign="middle">
<%Mathlet MSPBlock[ {$$fun},Factor[$$fun]]
%>
</td>
</tr>
</table>
<br>
<HR>
<img src=
"http://mathedu.h.kobe-u.ac.jp:8080/webMathematica/images/banners/webm-black-tiny2.gif"
alt="powered by webMathematica" border="0">
</BODY>
</HTML>
test_plot.msp
<HTML>
<HEAD>
<TITLE>グラフ表示</TITLE>
</HEAD>
<BODY BGCOLOR="#ffffff" >
表示結果<br>
<%Mathlet
MSPBlock[ {$$fun, $$x1},
MSPShow[ Plot[$$fun, {x,0,$$x1}]]] %>
<HR>
<img src=
"http://mathedu.h.kobe-u.ac.jp:8080/webMathematica/images/banners/webm-black-tiny2.gif"
alt="powered by webMathematica" border="0">
</BODY>
</HTML>
math.sql
DROP TABLE math;
DROP SEQUENCE math_id_seq;
CREATE TABLE test1 (
id serial primary key,
head text,
body text,
math text,
mathml text,
kind text,
value1 text,
value2 text,
value3 text,
value4 text,
value5 text
);
GRANT ALL ON math TO nobody;
GRANT ALL ON math_id_seq TO nobody;

```