

# 数学 Web サービスにおける グラフ表示サービスとその教育への応用

渡邊 巧

TAKUMI WATANABE

愛媛大学大学院理工学研究科

GRADUATE SCHOOL OF SCIENCE & ENGINEERING, EHIME UNIVERSITY \*

甲斐 博、野田 松太郎

HIROSHI KAI, MATU-TAROW NODA

愛媛大学工学部

GRADUATE SCHOOL OF COMPUTER SCIENCE, EHIME UNIVERSITY †

## 1 はじめに

数式処理・数値計算の工学や教育分野への応用について考える時、それらを利用した問題解決環境 (Problem Solving Environments:PSE)[2] の自動構築を行うことができるシステムが望まれる。従来の PSE の構築においては、システムに依存していたため、高コストの作業を必要としていた。また、拡張性に乏しいものであった。そこで、数式処理と数値計算を利用して問題を簡単に解くことや、強力なハードウェアとアルゴリズム能力を効果的に問題解決に生かすといった目的として、PSE 構築環境である Lupin[1] が提案された。Lupin は愛媛大学の我々の研究室の研究プロジェクトである。

Lupin は Web サービスの仕組みを利用することによって PSE を構築する。この Web サービスを数学分野に応用したものを特に数学 Web サービスと呼んでいる。本研究は数学 Web サービスにおけるグラフ表示についての研究である。

## 2 Lupin

Lupin は標準 Web 技術を基盤とし、Web ベースの PSE の構築を目的としているため特定のプラットフォームに依存しない。そして、拡張性が高い PSE 構築環境を目指している。現在、Lupin の研究は簡単な PSE の構築、機能の検証、評価をプロトタイプの実装によって行い、それらのプロトタイプの実用についての検討を行っている。

図 1 は Lupin のアーキテクチャを示している。この図のように Lupin のアーキテクチャは 3 つに分けることが出来る。ここでいう Lupin service とは数式処理や数値計算の能力を提供するものである。Lupin service generation mechanism は Lupin service の作成、配置を行い、Lupin discovery mechanism に Lupin

\*watanabe@hpc.cs.ehime-u.ac.jp

†kai@cs.ehime-u.ac.jp,noda@cs.ehime-u.ac.jp

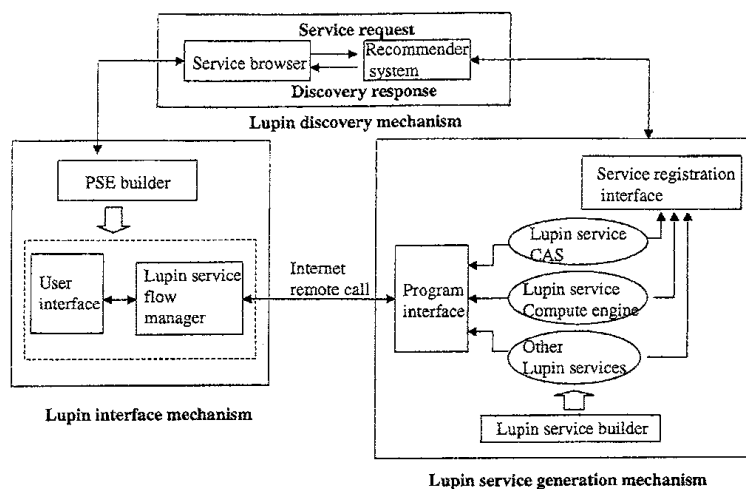


図 1: Lupin のアーキテクチャ

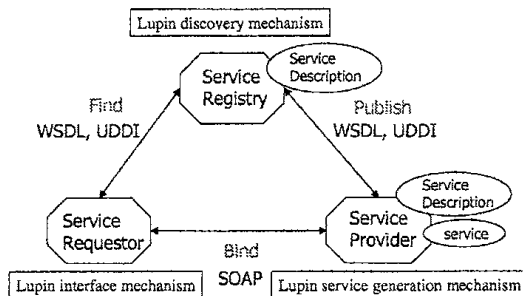


図 2: Web サービスによる Lupin の実装

service の情報を登録する。Lupin discovery mechanism は Lupin service の情報の検索・登録を行う。Lupin interface mechanism は PSE を構築するために Lupin discovery mechanism から必要な Lupin service の検索を行い、発見した Lupin service により PSE を構築する。

## 2.1 Web サービスによる Lupin の実装

Web サービスは http などの標準的なインターネットプロトコルと標準的なデータ記述方法である XML を用いるアーキテクチャである。このアーキテクチャを用いることにより、ユーザの課題や要求に対するソリューションを提供することが出来る。Web サービスは Service Provider、Service Registry、Service Requestor によって成り立つ。Service Provider はサービスの提供を行い、Service Registry はサービスの登録・管理・検索を行う。そして Service Requestor は Service Registry の検索によってサービスを発見し利用する。

Lupin のアーキテクチャに対して次のように Web サービスを対応させることにより、Lupin の実装を行っている。

- Lupin service generation mechanism を Service Provider に対応

- Lupin discovery mechanism を Service Registry に対応
- Lupin interface mechanism を Service Requestor に対応

これを図 2 に示す。SOAP・WSDL・UDDI は Web サービスにおける基本技術であり、それぞれ通信技術・サービス記述言語・Web サービスのレジストリ技術である。

### 3 数学 Web サービス

Lupin は Web サービスによって実装されているが、Web サービスの数学分野 (特に数値計算・数式処理) への応用を特に数学 Web サービスと呼んでいる。数学 Web サービスにおいて数学表現は MathML[3] や OpenMath[4] を用いている。

#### 3.1 MONET

数学 Web サービスに関連する研究として MONET(Mathematics on the Net)[5] がある。MONET は NAG, バス大学, ウェスタンオンタリオ大学などによるプロジェクトである。このプロジェクトは Web サービスを使って数学アルゴリズムを提供、利用する研究を行っている。MONET も同様に数学 Web サービスと呼んでおり、Lupin と同様の研究を行っている。

#### 3.2 実装されている数学 Web サービス

現在、Lupin によって実装されているサービスには次のようなものがある。

- 数値行列計算サービス
- 数式処理サービス

数値行列計算サービスは JLAPACK を利用するためのサービスである。JLAPACK[7] は LAPACK[6] を JAVA で利用できるようにしたものである。数式処理サービスは Risa/Asir の組み込み関数を利用することができ、さらに、Risa/Asir の自作プログラムを登録し利用することのできるサービスである。

実際に今までに実装された数学 Web サービスの実装を図 3 に示す。数式の入出力操作は、MathML の表示・入力支援ソフトウェアである WebEQ[8] を用いている (B)。まず最初に、利用したいサービスの選択を行う (A)。今回は多項式の因数分解を行うサービスである "Poly factorization" を選択する。次に数式を入力し (B)、"send" ボタンを押す (C)。すると、入力した数式をあらわす MathML がサービスに SOAP で送信される。サービスは受け取った MathML から問題を解決し、SOAP で解を返信する。そして、返信された解がクライアント画面に表示される (D)。実際に用いられた SOAP を図 4 に示す。四角で囲んだ部分が MathML の Content Markup が挿入されている部分である。

### 4 グラフ表示サービス

従来の Lupin における数学 Web サービスは計算を行うサービスに関する研究が中心であった。これは MONET も同様である。しかし、工学・教育等の分野への応用を考えると、結果の可視化・解析や関数の振る舞いの理解が必要になる。そこでグラフ表示サービスの実装の提案を行う。

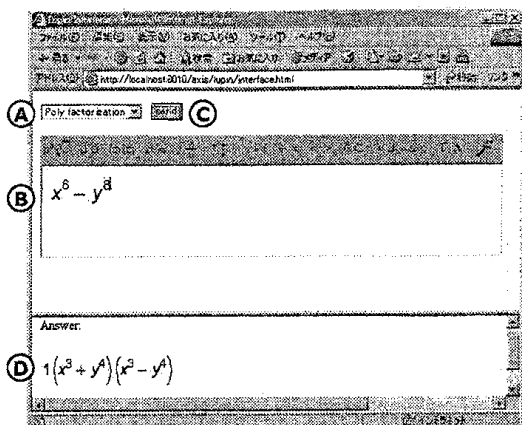


図 3: Lupin サービス実行画面

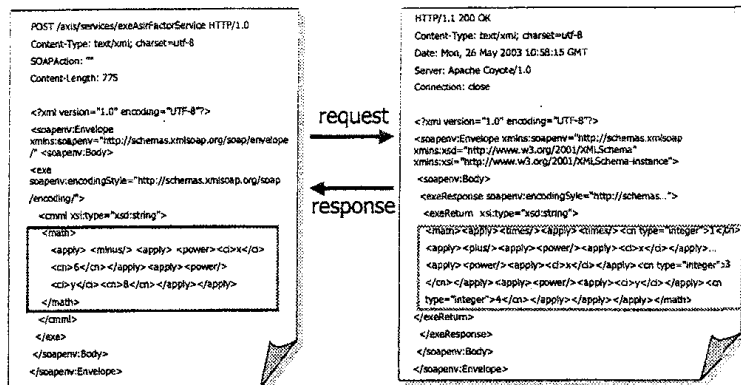


図 4: SOAP 通信の例

### 4.1 使用する技術

グラフ表示サービスは数式を入力するためのデータ形式として MathML の Content Markup を使い、グラフを表示するための画像形式として SVG(Scalable Vector Graphics)[9] を用いる。そしてグラフデータを作成するためのアルゴリズムとして Sign Weak Character[10] を用いる。Sign Weak Character は Risa/Asir に ifplot として実装されているアルゴリズムである。

SVG によりグラフ表示を行うことによって得られる利点として次のようなものが挙げられる。SVG はベクターグラフィックスであるので、一つのグラフを一つの図形であるとみなすことができ、それらの追加と削除を簡単に行うことが出来る。よって、グラフの重ね合わせや必要なグラフだけ表示するという処理を容易に行うことが出来る。さらに SVG は数式・グラフの表示範囲などのメタデータを格納することが出来る。このメタデータを利用して Sign Weak Character によるグラフの画像データからの精度を保った高速な拡大・縮小を行うことが出来る。今回、拡大を行うためにグラフ拡大サービスも実装している。

### 4.2 実装環境

表 1 の環境下で、グラフ表示サービスの実装を行った。グラフ表示サービスのクライアントは JSP/Servlet で実装を行っており、グラフ表示サービスは Java で実装している。Sign Weak Character を用いたグラフデータの作成における解の存在領域は、グラフ表示サービスから Risa/Asir を呼び出し計算させている。

表 1: 実装環境

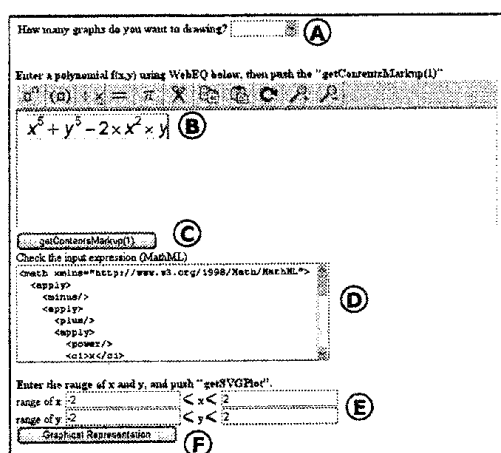
OS	Windows 2000 Professional
Java	Java2 SDK Version 1.4.2
servlet container	Apache Tomcat 5.0.28
SOAP Implementation	Apache Axis 1.1
Risa/Asir	Risa/Asir Version 20021003(Kobe Distribution)

### 4.3 実装

今回、一つの関数を入力してグラフを表示するクライアントと、二つの関数を入力し、それらの関数のグラフを重ね合わせて表示するクライアントを実装している。これらを利用し実装したグラフ表示サービス・グラフ拡大サービスを利用する例を以下に示す。

#### 動作例 1

図 5 は一つの関数のグラフを表示するクライアントの実装例である。まず、メニューによって表示するグラフの数を選擇する (A)。次に、WebEQ によって関数を入力した後 (B)、“getContentMarkup” ボタンを押す (C)。“getContentMarkup” ボタンを押すことにより、入力した関数の MathML(Content Markup) がフィールドに出力される (D)。最後にグラフの表示範囲を指定し (E)、“Graphical Representation” ボタンを押すと (F)、MathML(Content Markup) が SOAP でグラフ表示サービスに送信される。グラフ表示サービスは送信された関数からグラフを表す SVG(以下、グラフ SVG と呼ぶ) を作成し、クライアントに返信する。その結果は図 6 のように表示される。この結果の画面において“zoom” ボタンを押すと (G)、現在表示しているグラフ SVG がグラフ拡大サービスに SOAP で送信される。そして拡大されたグラフ SVG が返信され、図 7 のように表示される。



How many graphs do you want to drawing? (A)

Enter a polynomial f(x,y) using WebEQ below, then push the "getContentMarkup()"

$x^2 + y^2 - 2x^2 \cdot y$  (B)

getContentMarkup() (C)

Check the input expression (MathML)

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <apply>
    <minus/>
    <apply>
      <plus/>
      <apply>
        <power/>
      </apply>
    </apply>
  </math>
```

Enter the range of x and y, and push "getSVGPlot".

range of x: -2 <x> 2 (E)

range of y: -2 <y> 2 (E)

Graphical Representation (F)

図 5: 一関数入力画面

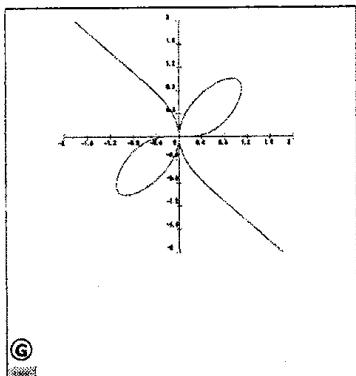


図 6: 一関数グラフ

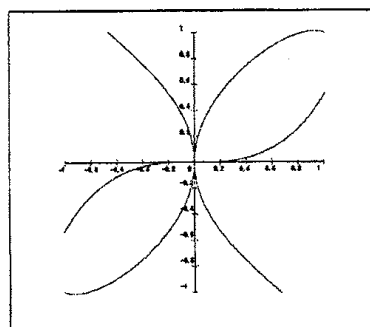


図 7: 拡大グラフ

動作例 2

図 8 は二つの関数を入力し、それらの関数のグラフを重ね合わせて表示するクライアントの実装例である。先ほどの一つの関数のグラフの場合と同様に、それぞれの関数を入力、そしてグラフの表示範囲を指定する。そして”Graphical Representation”ボタンを押し、グラフ表示サービスにそれぞれの関数の MathML(Content Markup) を別々に SOAP によって送信する。グラフ表示サービスは MathML(Content Markup) が送信される毎にグラフ SVG を作成する。クライアントは返信されたそれぞれのグラフ SVG を重ね合わせて表示するように変換し、結果を図 9 のように表示する。この結果の画面において”graph of first expression” ボタンを押すと (H)、最初に入力した関数のグラフのみが表示される。このときクライアントでは重ね合わせのグラフ SVG から最初に入力した関数のグラフ SVG のみを取り出して表示している。”graph of second expression” ボタンにおいても同様な方法で二つ目に入力した関数のグラフ SVG のみを表示する (I)。この結果はそれぞれ図 10 と図 11 に載せる。”zoom” ボタンを押すと (K)、表示しているグラフ SVG をグラフ拡大サービスに送信し、得られた拡大したグラフ SVG を表示する。

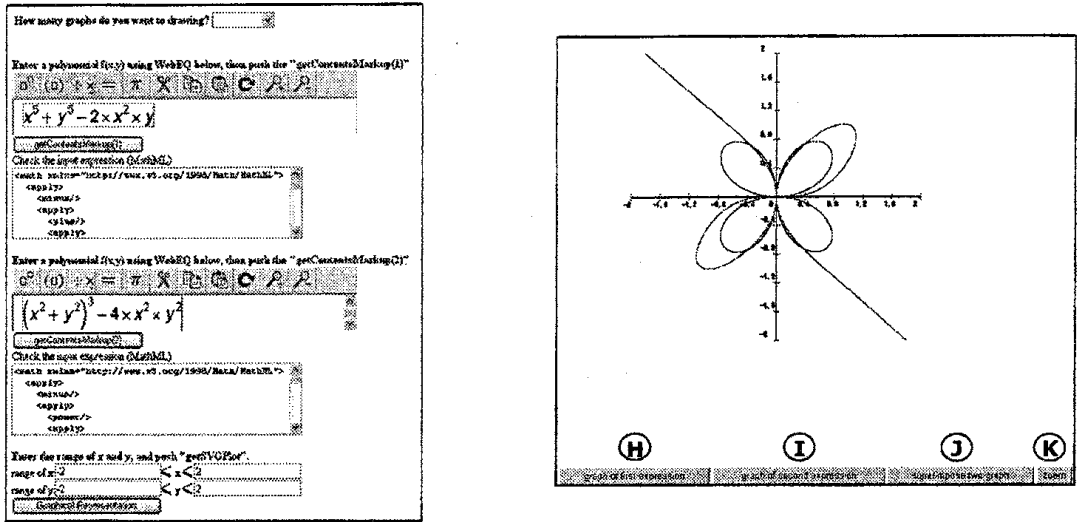


図 9: 二関数グラフ

図 8: 二関数入力画面

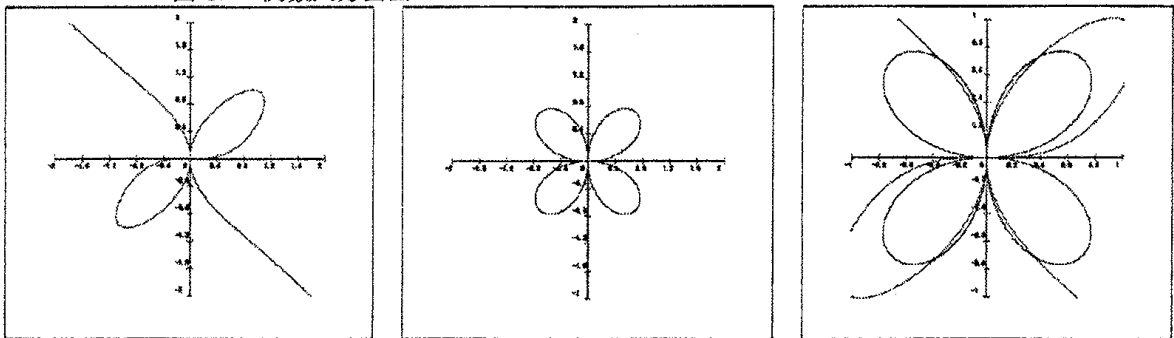


図 10: 最初に入力した関数のグラフ 図 11: 後に入力した関数のグラフ

図 12: 拡大グラフ

## 5 グラフ表示サービスの教育への応用

グラフ表示サービスを教育分野に応用することを考えるにあたり、アメリカのケント州立大学で研究・開発が行われている WME(Web-based Mathematics Education)[13] と呼ばれる数学教育 Web ページの作成用フレームワークに注目した。WME は独自の XML ベースの数学教育ページ作成用言語である MeML をコアとしている。そして WME サービスと呼ばれるサービスを利用することによって、動的なコンテンツの作成を行っている。

### 5.1 WME

WME のアーキテクチャは図 13 のようになる。まず Web ブラウザは MeML が格納されている Web サーバにアクセスし、MeML を読み込む。そして MeML Plug-in が MeML を XSL や DOM などを用いて Web ブラウザで表示できる形に変換する。この時、WME サーバによって計算された結果を用いたコンテンツを作成する場合や、グラフを表示するなどの複雑な処理を行う場合に WME サーバを用いる。

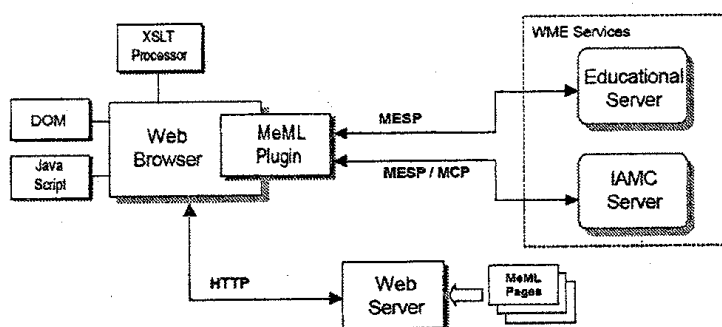


図 13: WME のアーキテクチャ

#### 5.1.1 MeML

MeML には次ようなようなタグが存在する。

- Content elements - 数学の知識をあらわす要素 <expression>, <variable>, <concept>, <equation>, <identity>, <skill>, <terminology>, <diagram>, <theorem>
- Education elements - 教育オブジェクトのための要素 <test>, <example>, <exercise>, <homework>, <lesson>, <assessment>, <diagnosis>, <remediation>
- Organization elements - コンテンツの構成のための要素 <guide>, <syllabus>, <roster>, <summary>, <hint>, <abstract>
- Computation elements - 動的コンテンツのための要素 <computation>, <mathgraph>
- Interaction elements - ユーザの入力のための要素 <interaction>, <userinput>

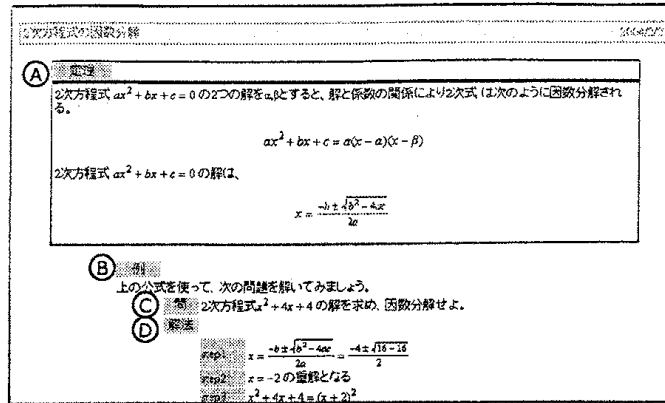


図 14: MeML ページ

式を表す expression タグや概念を表す concept タグ、レッスンを表す lesson タグや例を表す example タグなど様々な数学教育ページを作成するためのタグがある。そして、前述した WME サーバを用いたコンテンツを作成するためのタグも用意されている。WME サーバによる計算結果を利用したコンテンツを作成するときには computation タグを用い、WME サーバを用いてグラフを表示するためには mathgraph タグを用いる。

実際に MeML を元で作成した Web ページを図 14 に載せる。定理を示す部分は theorem タグ囲んでおり (A)、例は example タグで囲んでいる (B)。例についての記述部分の中でも、問題を記述する部分は problem タグで囲み (C)、そして、解法を記述する部分は skill タグで囲んでいる (D)。

### 5.2 WME の拡張

本研究では、教育分野への応用として WME の Lupin による拡張を行った。WME は WME サーバとの通信に独自の通信プロトコルを用いている。そこで、より一般的な通信プロトコルである SOAP を追加することにより拡張性を上げ、数学 Web サービスを用いることが出来るようにする。

前述した WME のアーキテクチャは図 16 のように二つに分けて考えることが出来る。左側をサービスを利用する側である Lupin interface mechanism とみなし、そして、右側をサービスを提供する側である Lupin service generation mechanism とみなす。このことから WME アーキテクチャを Web サービスを用いて実装することが出来ると考えられる。

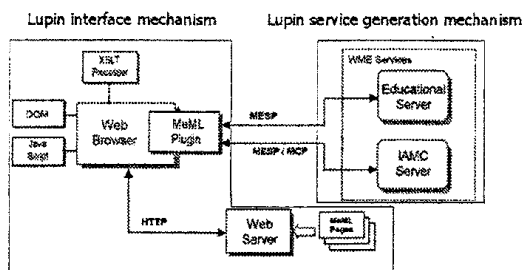


図 15: WME と Lupin の関係

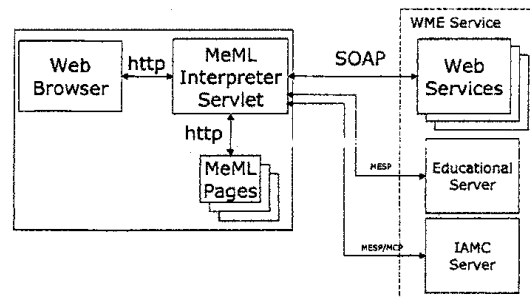


図 16: Web サービスを用いた WME の実装



図 16 は今回実装を行った Web サービスを用いた WME アーキテクチャである。このアーキテクチャでは MeML プラグインの機能を MeML Interpreter Servlet として外部の機能としている。これはブラウザは表示、MeML Interpreter は MeML からブラウザへ表示できる形式への変換というように、機能を分散させ、ブラウザへの負担を減らすことを目的としている。MeML Interpreter は MeML Plug-in の機能と同様に MeML をブラウザが表示できる形に変更する。そして、コンテンツを作成する時に WME サービスのほかに Web サービスを用いてコンテンツを作成することが出来る。

MeML をグラフ表示サービスへ対応させるために mathgraph タグの下に webservice タグを追加するという手段をとった。webservice 要素は Web サービスの URL を記述するための url 属性、サービス名を記述するための operation 属性を持っている。そして webservice タグの下に SOAP によって通信する XML を挿入する。実際の webservice タグの例を図 17 に載せる。四角に囲まれている部分は上から順に関数を表す MathML、x 座標の範囲、y 座標の範囲を指定する XML である。このタグが含まれた MeML は MeML Interpreter Servlet による変換の途中で、SOAP によりグラフ表示サービスに送信され、結果を変換後の Web ページにおいてグラフを表示することが出来る。webservice タグを挿入した MeML を MeML Interpreter Servlet によって変換した結果を図 18 に載せる。

```
<mathgraph>
<ns1:webservice url="http://10.0.0.234:9090/lupin/SVGGraph"
xmlns:ns1=http://hpc.cs.ehime-u.ac.jp/lupin/SVGGraph/
operation="svgplot" >
<ns1:function>
<math xmlns="http://www.w3.org/1998/Math/MathML">
<apply><minus/>
<apply><minus/><apply><minus/><ci>y</ci>
<apply><power/><ci>x</ci><cn>2</cn></apply></apply>
<apply><times/><cn>4</cn><ci>x</ci></apply></apply>
<cn>4</cn>
</math>
</ns1:function>
<ns1:xmax>4</ns1:xmax>
<ns1:xmin>-6</ns1:xmin>
<ns1:ymax>8</ns1:ymax>
<ns1:ymin>-2</ns1:ymin>
</ns1:webservice>
</mathgraph>
```

図 17: 拡張した mathgraph タグの例

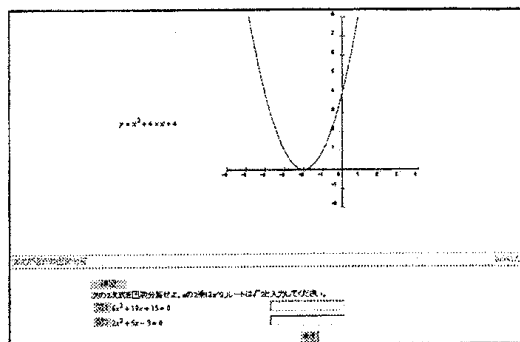


図 18: グラフ表示サービスを利用した MeML ページ

## 6 まとめ

本研究では、まず、グラフ表示サービスとそれを利用するためのクライアントの実装を行った。その実行結果より MathML を数式の入力とし、SVG をグラフ表示に用いることが出来ることが分かった。このことからグラフ SVG の拡大前と同精度における拡大、そして、再計算を行わないグラフの重ねあわせと選択したグラフの抽出が可能となった。

教育への応用として MeML Interpreter の実装を行うことにより、MeML を Web ブラウザーで表示可能な形式への変換を行うことが出来た。そして、Web サービスの利用を可能としたため、グラフ表示サービスを利用することが可能となった。この実装により、WME の拡張性はあがったと考えられる。

## 参 考 文 献

- [1] K. Li, M. Sakai, Y. Morizane, M. Kono and M. -T.Noda, Lupin:Towards the Framework of Web-based Problem Solving Environments , InProc. ATCM'03, Taiwan, pp.276-285,2003

- [2] Eliaa N. Houstis and John R. Rice, On the Future of Problem Solving Environments, March 2000.<http://www.cs.purdue.edu/people/jrr>
- [3] W3C, MathML, <http://www.w3.org/Math/>
- [4] OpenMath, The OpenMath website, <http://www.openmath.org/>
- [5] The MONET Consortium, MONET Home page, <http://monet.nag.co.uk/>
- [6] LAPACK, <http://www.netlib.org/lapack/>
- [7] JLAPACK, <http://icl.cs.utk.edu/f2j/>
- [8] WebEQ, <http://www.dessci.com/en/>
- [9] W3C, Scalable Vector Graphics(SVG), <http://www.w3.org/Graphics/SVG/>
- [10] 齋藤友克, Displaying Zero of Mathematical Equations, 平賀工業社, 2000
- [11] Apache Tomcat, <http://jakarta.apache.org/tomcat/>
- [12] Apache Axis, <http://ws.apache.org/axis/>
- [13] Web-based Mathematics Education(WME), <http://wme.cs.kent.edu/>