

Wiki や blog でも数式を — BrEdiMa を使って —

電気通信大学電気通信学研究科 中野 泰人¹ (NAKANO Yasuhito)
森光 大輔 (MORIMITSU Daisuke)
村尾 裕一 (MURAO Hirokazu)
Univ. Electro-Communications

1 はじめに

BrEdiMa とは Web ブラウザ上で動作する, GUI による数式の編集を行うための Web アプリケーションで, JavaScript で書かれているので既存の Web ページに簡単な変更をするだけで GUI 数式エディタを追加することができる. その名前は Browser, Edit 及び Math の一部を組み合わせたものである. BrEdiMa には, 公式サイト [4] でも述べられているように, 次のような特徴がある.

- ほとんどのサーバですぐに設置できる.
- ほとんどの Web ブラウザで準備なしに動く: JavaScript のみで記述されていて, 表示も HTML と画像のみを使用している. Internet Explorer, Mozilla Firefox をはじめとした多くの Web ブラウザで動作可能.
- シンプル: 使用できる機能を高校の学習内容までに絞りシンプルにまとめており, 数学が専門でない人でも迷うことなく利用することができる.
- MathML [1] にも対応
- 応用範囲が広い: API も用意されており, Web ページ制作者が簡単な JavaScript を記述するだけで様々な用途に利用可能.

以上の特徴は完成したソフトウェアのセールスポイントである. 開発にあたって掲げた目標は, (i) 手軽に使えて, カジュアルな用途にも利用可能, (ii) プラグイン等を要しない, (iii) AJAX などに合致するなどだが, ほぼ達成されている. 数式を編集するためのソフトウェアは, W3C の Math のホームページ [2] に列挙されているように MathML の応用例を中心として多数存在する². 中でも BrEdiMa は JavaScript だけで記述されていることが大きな特徴となっている. その後, 同様のソフトウェアとして MathEdit[7] が出現している.

¹本稿の研究開発を行った時点での所属. 2008 年 4 月以降は (株) ヤフー (Yahoo Japan Corp.)

²MathML の Internet や Web での利用状況に関する報告などもある.

上の目標はあくまでも開発者の視点から利用者層を想定して掲げた利点に過ぎず、プログラム開発者としての力点は、プログラミングの題材として（Google Maps と AJAX とも関連して）JavaScript 利用の見直しや最新の Web 技術の利用に置かれているのが正直なところであろう [3]。そもそも、今日、インターネット或は Web ブラウザの利用者の大多数はコンピュータ技術とは無縁の人達であり、そのような利用者がブラウザを使って文章を作成する場面と言えはブログ（WebLog, blog, ...）や Wiki（人によっては Wikipedia）が主なものであろう。上でも触れた多様なソフトウェアがそのような場面で活用できるのかは不明だが、当初からブラウザ上での利用のために作られた BrEdiMa はそのような場面への活用は（上の特徴にも挙げているとおり）とても簡単にできる。実際、BrEdiMa の最初の版では、wiki クローンのひとつである Hiki 用のプラグインが JavaScript の数行のプログラムを追加して開発されている。

一方最近では、ブラウザの利用者が JavaScript プログラムを利用者側で用意し、ブラウザの機能を拡張して用いるという使い方が流行りつつある。従来 JavaScript プログラムは Web ページの設置者の意向で HTML 等による静的な記述を補足する形で使われていたが、この機能拡張としての利用形態はブラウザ利用者が能動的に導入するという点で従来と大きく異なる。

blog や wiki が文章を書く場所つまり紙とすれば、数式記述の道具である BrEdiMa は鉛筆と消しゴムである。鉛筆と消しゴムは、紙とは独立して書く側の人が好みで自由に選ばばよい。

BrEdiMa の最新版 [6] は、ブラウザの機能を拡張するスクリプトとして導入可能な形に作り変えられている。鉛筆と消しゴムとして、色々な Web ページに利用者自身で自由に使い回せるようにするためである。本稿では、その利用法を利用者向けに説明する。次節 (§ 2) ではブラウザの機能拡張の方法を説明し、拡張して利用した場合のスクリーンショットを示す。実際の利用に至るまでの導入手順は付録に示す。§ 3 では、拡張機能を記述するユーザスクリプトの内容を説明する。本稿で示す以外の blog や wiki で利用する場合に新たに記述ためのマニュアルとなる。BrEdiMa 本体の数式編集の利用法については、過去の本研究集会でも紹介してきたし、最初の開発サイト³に試用やチュートリアルが用意されているのでそれらを参照されたい。

2 導入・利用の方法

BrEdiMa は JavaScript で書かれた単一のプログラムである⁴。BrEdiMa を用いるには、基本的に、このプログラム（の単一の .js ファイル）をブラウザに導入して起動する。その方法には、

- Web ページの提供者（管理者）にとって、

³個人サイトなのでここには明記しない。[3] のリンクから到達可能である。

⁴但し、アイコンの画像群と、(ブラウザが MathML 未対応の場合に) 式を画像に変換する mimeTeX[14] の CGI と、2つの外部のリソースを用いる。

- 只の Web ページの閲覧者として

と、それぞれの場合に対応して2種類ある。

前者は、旧来から用いられている方法で、BrEdiMaのJavaScriptファイルを然るべきサーバー上に置いておき、そのファイルを引用しているWebページの閲覧を契機としてWebページのダウンロードと連動してJavaScriptプログラムがブラウザに導入される。この旧来の方法による導入・設置法については、概要紹介などとともに[4]の「設置方法」のページに詳しい。

後者は、ブラウザの利用者（つまり、Webページの閲覧者）が、自らJavaScriptプログラムを入手してブラウザの拡張機能として利用するという方法である。ここ数年でJavaScriptによるブラウザ上でのプログラミングが盛んになり、ユーザスクリプトと呼ばれる、サーバー側ではなくブラウザ利用者が用意したJavaScriptプログラムを実行する環境が整備されるようになった。BrEdiMa開発の最新の成果はそのようなユーザスクリプトとしても利用可能な形への改訂[6]であり、上述のような利用形態を想定している。

2.1 ブラウザの利用者による機能拡張

Google Mapsとそこで用いられたAjaxというブラウザ上からの非同期通信を行う技術が注目を集め、JavaScriptを用いたブラウザ上で動作するプログラムの開発が脚光を浴びるようになると同時に、ブラウザを機能拡張するための一般的な方法（FirefoxのAdd-on [8]）、ブラウザ上のデバッグ環境（同じくFirebug [11]）や、手軽にブラウザの動作に変更を加えるためのユーザスクリプトの実行環境（同じくGreasemonkey [9]）などが整備された。現在、Greasemonkey用の（ユーザ）スクリプトの開発が盛んであり、多種多様なものが用意・公開されている⁵。GreasemonkeyはFirefoxのadd-onのひとつとして開発されたもので、ある形式や制限[10]に従ったJavaScriptで書かれたユーザスクリプトを実行するための汎用的な環境を提供する。Greasemonkeyの同じあるいは同等の環境をInternet Explorer (IE)上に構築するためのGreasemonkey for IE [12]、Trixie [13]なども開発されているが、互換性等の詳細は不明である。また、IEで機能拡張する環境には、他にもTurnabout、GreasemonkIEなどがある。

JavaScriptでは、ブラウザ上にダウンロードされたWebページのDOM木構造そのものやフォームへの入力データを操作・編集したり、キーボードおよびマウスの操作状況を検知することができる。従来は、そうした機能を用いてWebページの表現を補足するプログラムをサーバー側に準備していたわけだが、Greasemonkey等による機能拡張では汎用的な機能のJavaScriptプログラムをWebページとは独立に開発・準備し、どのページで用いるかを利用者が選択して用いる。拡張機能としては、表示法を変更や拡張したり、入力の補助や編集を行うといった例が多い。数式の記述法は、特殊な記号も含めた様々な記号を大きさを変化させながら1次元的に配置するものであり、BrEdiMaは筆記に即した数式の2次元入力を1次元で並べた文字列や単一の画像に変換する。数

⁵利用にあたっては、トロイの木馬やウイルスに注意が必要である

式の Web ページでの扱い方が十分に固まっていない現状において、数式の入力・編集の機能はフロントエンドの機能へと集約してしまい、blog や wiki 等の各種の Web ページの形式に対しては柔軟に対応できるような形に整備することは、その機能を固有のソフトウェアに作り込むよりは将来性も見込め、理想的な形のひとつといえることができるだろう。

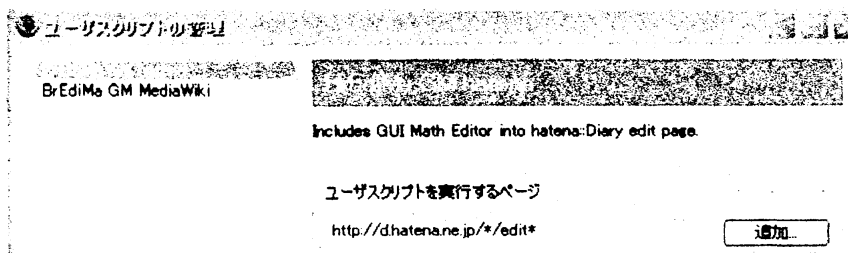
BrEdiMa の最新版は、このような考えに基づき、Greasemonkey のユーザスクリプトとしてすぐに利用できるように改訂されている。実際、その版を用いて blog のひとつであるはてなダイアリー用と、Wiki クローンについては FreeStyleWiki 用と Wikipedia で用いられている MediaWiki 用の 3 種類のユーザスクリプトがこれまでに作られている。

本節の以降では、実際の使用例（はてなダイアリーと FreeStyleWiki）をスクリーンショットを用いて示す。利用にあたっては、何段階かのソフトウェアの導入と設定が必要だが、その詳細は付録 A で説明する。ユーザスクリプトの構成については、次節 § 3 で説明する。

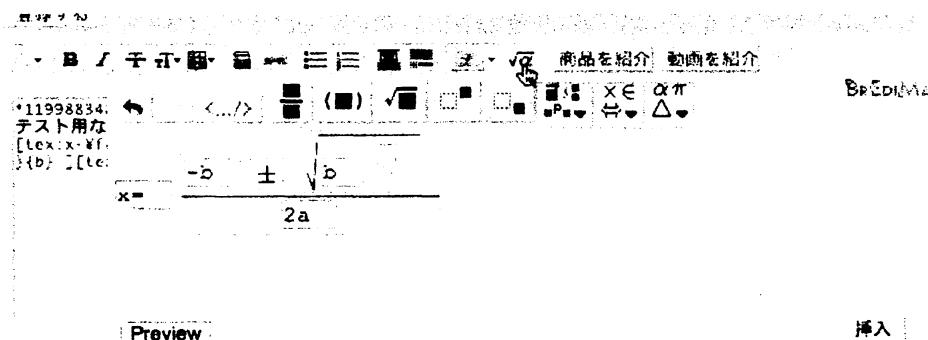
2.2 使用例

はてなダイアリーの場合

まず、ブラウザとしては Firefox を用い、予め付録 A に示す方法に従って Greasemonkey とはてなダイアリー用のユーザスクリプトを導入（本来は、および設定）しておく必要がある。導入済みかどうかは、ブラウザの右下の猿の顔のアイコンを右クリックし（あるいは、「ツール」→「Greasemonkey」から）「ユーザスクリプトの管理」を選択すると現れるウィンドウの一覧の中に、「BrEdiMa GM hatena」があるかで確認できる。併せて、Greasemonkey が有効（猿の画像が灰色でなく茶色）になっているかも確認する。



はてなダイアリーのページ (<http://d.hatena.ne.jp/>) にアクセスして編集を開始すると、BrEdiMa を起動するボタン ($\sqrt{\alpha}$ のアイコン) が増えている (下図)。



そのボタンをクリックすると、BrEdiMaが起動（して数式の入力モードが開始）する。入力・編集が完了したら「挿入」をクリックすれば、数式がはてなダイアリーのページで扱える形式に変換されて入力領域に挿入される。

FreeStyle Wiki の場合

Wiki クローンの一種である FreeStyle Wiki (略称 FSWiki. <http://fswiki.org/>) は、少なくとも国内では人気のある wiki のひとつで、様々なサイトと運用されている。FSwiki で BrEdiMa を用いる場合、はてなダイアリーの場合と同様にして適切なユーザスクリプト (BrEdiMa GM fswiki) が導入されているかを確認するだけでなく、「ユーザスクリプトの管理」において当該 FSwiki サイトが「ユーザスクリプトを実行するページ」に当てはまるか (実行するページとして指定されている文字列のパターンに当てはまるか) を確認する必要がある。当てはまらない場合は、URL を適宜変更・追加する (A.3)。ユーザスクリプトが有効になれば、編集画面にした時にはてなダイアリーの場合と同様、 $\sqrt{\alpha}$ の BrEdiMa 起動のボタンが現れる。図 1 は、このボタンをクリックして数式を入力中の例である。

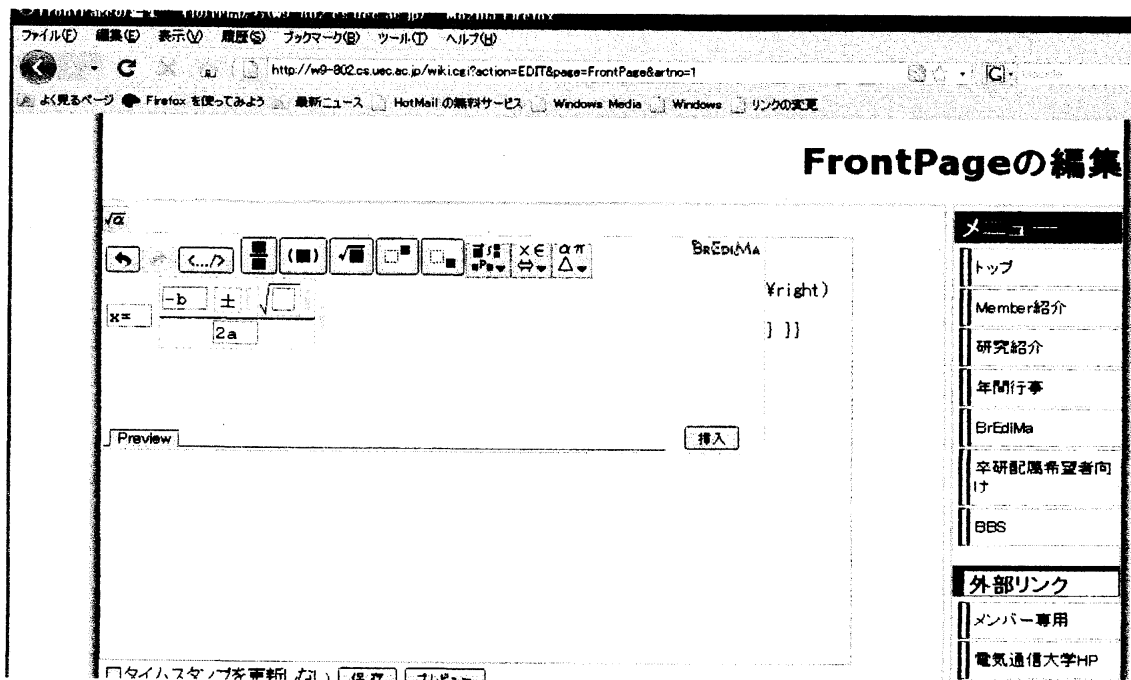
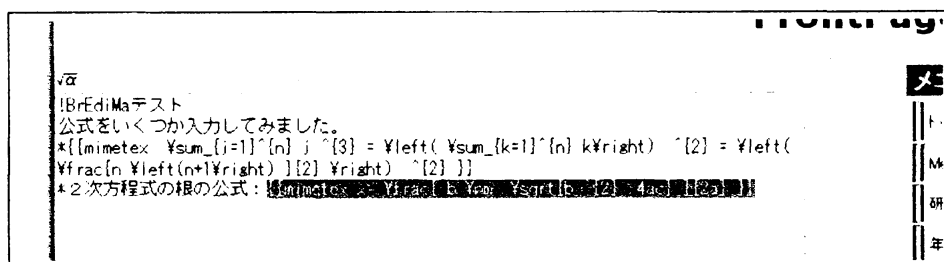
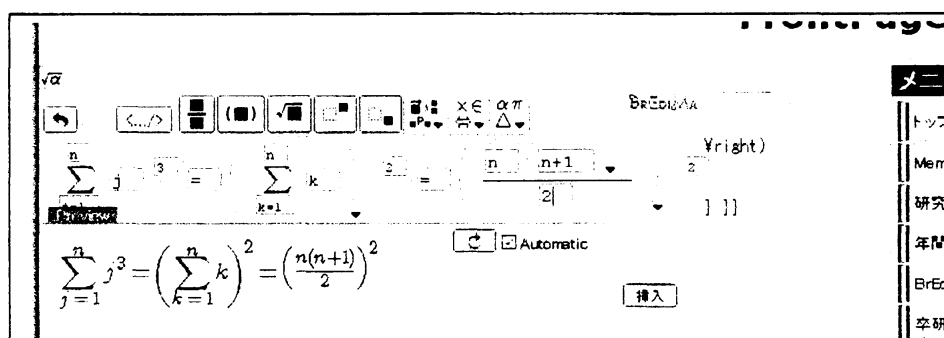


図 1: FreeStyle Wiki での利用：BrEdiMa を起動したところ

式の入力・編集が完了し「挿入」ボタンをクリックすると、FSwiki で扱える形式 (FSwiki では、LaTeX 形式の式を mimeTeX プラグインがレンダリング) の数式 (下図の反転部分) が挿入される。



上のどちらのユーザスクリプトの場合も、BrEdiMaでの数式の入力・編集にプレビューが可能であり、「Automatic」をチェックしておくとそのプレビューは入力・編集に応じて変化していく。



3 ユーザスクリプトの作成

BlogやWikiでは、利用者がサーバー側のことをあまり気にせずに、元のWebページを編集することができる。その延長でサーバー側の設定を気にすることなく、簡単に数式を扱えるようにするために、BrEdiMaをGreasemonkeyのユーザスクリプトで利用する方法を提供する。現行のBrEdiMaはそのような利用が可能ないように改訂されている。

本節では、BrEdiMaを利用するためのユーザスクリプトの作成方法を説明する。まず事前準備として、次の3点を確認する必要がある

- 目的のサイト (blog や Wiki) で数式は扱えるのか?
mimetex を用いるためのプラグイン等、数式を画像に変換 (あるいは、MathML をレンダリング) する機構が存在するかを確認する
- 扱える場合、数式はどのように表現するのか?
Blog や Wiki の Web ページ中での扱いに合う出力形式を調べる (LaTeX あるいは MathML)
- また、数式データはページのどの部分に対して追加するのか?
これは、Web ページのソースを見れば大抵はわかる
- 動作対象の URL: ユーザスクリプトを動作させるページの URL が、どのような文字列になるか

```

①: // ==UserScript==
②: // @name          BrEdiMa GM template
③: // @namespace     http://bredima.sourceforge.jp/
④: // @description   Includes GUI Math Editor into ??? edit page.
/* 本スクリプトを実行するページの URL */
⑤: // @include      http://d.hatena.ne.jp/*/edit*
⑥: // ==/UserScript==

/* Copyright の記述 */

⑦: Bredima= ... /* (BrEdiMa 本体の定義等) */
⑧: window.addEventListener(
⑨:   'load',
⑩:   function() {
// 適用可能なページかを判定
⑪:   if(!document.getElementById('edit-buttons')) return;
// BrEdiMa を起動するボタンになるイメージオブジェクトの作成
⑫:   var button = document.createElement('img');
⑬:   button.src='http://bredima.sourceforge.jp/pub/pubimg/
                                     gm_hatena.png';
⑭:   button.style.marginRight = '5px';
// そのボタンを当該ページ内に設置
⑮:   document.getElementById('edit-buttons').appendChild(button);

// mimetex のサーバの URL と BrEdiMa が用いる画像の存在する URL の設定
⑯:   Bredima.setConfig('uri_mimetex', '/cgi-bin/mimetex.cgi');
⑰:   Bredima.setConfig('uri_img', 'http://bredima.sourceforge.jp/
                                     pub/img/');
⑱:   Bredima.setConfig('isGM', true);
⑲:   // BrEdiMa オブジェクトをボタンに関連付けて生成
⑳:   var bd = new Bredima(button, 'float');
㉑:   bd.setConfig('use_button', true);
㉒:   // 入力された数式データの当該ページへの挿入法：場所と形式の指定
㉓:   bd.onsubmit = function() {
㉔:     var form = document.forms.namedItem('edit');
㉕:     var area = form.elements.namedItem('body');
㉖:     Bredima.insertTo(area, '[tex:' + bd.toLatex() + ']');
㉗:   }
㉘: },
㉙: false);

```

図 2: はてなダイアリーで BrEdiMa を利用するための Greasemonkey ユーザスクリプト

これらの情報を元にユーザスクリプトを記述する。図 2 は はてなダイアリー用のユーザスクリプトである。これに基づいて説明する。

先ずスクリプトは 次の (A)~(C) の 3 部分から成り、主な内容は次のとおり。

(A) 当該スクリプトに関するメタデータ (①~⑥)

- 動作対象の URL の指定 (⑤)

(B) BrEdiMa プログラム本体 (⑦) — 通例に従い (?) 長大なプログラムも 1 行で書かれている,

(C) 当該スクリプトが実行される Web ページがブラウザ上にロードされた時点で行う各種設定の記述 (⑧~⑳)

- BrEdiMa 起動ボタンのページ内への設置 (⑫-⑬及び⑭-⑮)
- BrEdiMa が利用する外部リソースの指定 (⑯-⑰)
- 入力された数式データの当該ページへの挿入法の指定 (㉔-㉖)

これらの処理をひとつの関数内に記述し、⑧-⑨で、'load' というイベントが発生した時点 (ページのブラウザへの読み込み完了) でその関数が呼ばれるように設定している。

動作対象の URL の指定 (⑤)

- スクリプトのメタデータとして、当該スクリプトを実行し動作の対象とするページの URL の文字列を @include で指定する。
- 逆に、ある種のページに対して動作しないようにする「@exclude」という指定の仕方もある。
- ここでの記述が当該スクリプトのインストール直後の初期値を与えるが、インストール後のブラウザへの指定は、ブラウザの [ツール] → [Greasemonkey] → [ユーザスクリプトの管理] で追加・変更が可能 (付録 A.3 に示す)。
- 総称文字 * で任意の文字列を表すことが可能。
- BrEdiMa を wiki の編集ページに対し動作させるために、URL の文字列の一部として FreeStyleWiki の場合は「action=EDIT」と、MediaWiki の場合は「action=edit」と指定する。MediaWiki で運用される日本語 Wikipedia では、編集ページは
「<http://ja.wikipedia.org/w/index.php?title=...&action=edit&...>」
となるので
「http://*/w/*&action=edit*」のように指定すれば良いだろう。

BrEdiMa 起動ボタンのページ内への設置 (⑫~⑮)

清書システムでは普通のテキストを入力・編集するモードと数式を入力・編集するモードの 2 つのモードを持つのが普通である。Blog や Wiki で編集を行う場合も、この 2 つのモードを持たせるのが自然だが、後者の役を BrEdiMa に担わせる。そ

れには、編集画面において、BrEdiMaを起動してテキストとは別の入力モードで数式の入力・編集を行わせるが、BrEdiMaモードを開始するためのアイコンを設置する必要がある。

はてなダイアリーの編集画面には、文字フォントの指定や箇条書きのための入力支援ツールが設置されているが、ここにBrEdiMaを起動するためのボタンを追加する。

- ⑪ : 入力支援ツールのアイコン群は id 名が edit-buttons の要素中に置かれているので、先ず、そのタグの存在を確認する。
- ⑫-⑭ : BrEdiMa 起動ボタンのアイコンとなる img タグを生成する。画像ファイルそのものは ⑬の右辺の URL から取得するが、他にも [5] の http://w9-802.cs.uec.ac.jp/img/bredima_icon.png から入手可能。
- ⑭は画像の位置調整。
- ⑮ : このアイコンの画像を、id 名 edit-buttons のタグ中の子の一員として末尾に追加。

Wiki 用の場合：編集画面における入力領域 (textarea) は、DOM ツリーでは (ほぼ間違いなく) 先頭のフォーム (document.forms[0] で表される) である。⑪の判定とボタンの追加は、このオブジェクトに対して行うことにする。この方法は wiki の設置者が変更したとしても対応可能なので、特定の id 名の要素を指定するのに比べ、より一般的な方法となる。

⑪相当 :

```
if(!document.forms[0]) return;
```

- ⑭-⑮ : 設置の方法は、入力フォームの先頭にアイコンのオブジェクトを改行タグを挟んで追加する。

```
var form = document.forms[0];
form.insertBefore(document.createElement('br'),
                    form.firstChild);
form.insertBefore(button,form.firstChild);
```

BrEdiMa が利用する外部リソースの設定 (⑯~⑰)

(ブラウザが MathML に対応していない場合に必要) LaTeX で書かれた数式を画像に変換する mimeTeX の CGI の URL (⑯) と、特殊文字等のアイコンの画像の存在する URL (⑰) を指定する。**はてなダイアリー**用の⑯では同サイトが提供する CGI を利用するようにしている。一般には、次のように指定する (これらの URL のサービスの継続性の保証はない)。

```
Bredima.setConfig('uri_mimetex',
                  'http://w9-802.cs.uec.ac.jp/mimetex.cgi');
Bredima.setConfig('uri_img', 'http://w9-802.cs.uec.ac.jp/img/');
```

⑱の isGM に対する設定は、Greasemonkey で使用することを指定。

⑳～㉑ :

- Bredima オブジェクトのインスタンスを生成 (㉑)。第一引数には、編集領域の挿入先の div 要素の DOM オブジェクトを指定。第二引数で float を指定している場合は編集領域を表示させる際にクリックする DOM オブジェクトを指定。
- 「挿入」ボタンの表示 (㉑) : BrEdiMa で作成した数式を入力領域に挿入することを指定するボタンを編集領域の右下に表示することを指定。
- その他の設定の詳細は [4] の BrEdiMa リファレンス参照。

数式データの挿入法の指定 (㉒～㉔)

BrEdiMa での入力・編集作業が終わり「挿入」ボタンがクリックされた時の処理を、仕掛ける無名関数のボディに記述して指定する。数式を textarea 中のテキストの一部として埋め込むが、DOM ツリーのどの要素に追加するかとどのような文字列に変換して埋め込むかは blog や wiki に依存する。はてなダイアリーの場合、edit という名前のフォーム (㉒) 中の、body という名前の textarea (㉓) に、

```
[tex: LaTeX形式の数式]
```

という形にして挿入する (㉔)。

Wiki 用への改訂 :

- 挿入場所の取得 : ㉒-㉓に相当する処理は、ここでは、フォーム内の textarea を (ループでスキャンして) 探し出すことにする。これは、Wiki では基本的に編集ページのフォームに textarea が一つだけ用意され、それが送信データになることを仮定した。これで上手く動作しない場合は、textarea に id を指定すればよい。

```
var form = document.forms[0]; var area;
for(var i=0; i<=document.forms[0].elements.length; i++) {
    if (document.forms[0].elements[i].type=="textarea") {
        area = form.elements[i];
        break;
    }
}
```

- 数式データの挿入時の形式の指定：Web 上で数式を扱うための規格・形式は MathML にほぼ固まったとは言え、ブラウザの対応が進んでいないため、blog や wiki での数式の利用は拡張機能として提供され、形式もまちまちである。そのため⑳において挿入する数式の様式は、利用する blog や wiki の各々に適合させる必要がある。

blog/wiki	形 式
はてなダイアリー	[tex: LaTeX形式]
FSwiki	{{mimetex LaTeX形式}}
Mediawiki	$LaTeX形式$

- BrEdiMa で再編集可能なのは JSON(JavaScript Object Notation)[15] だが、wiki クローンでは普通は扱えない → 再編集を実現するには既存 wiki への修正が必要

4 おわりに

本文では特に触れていないが、数式を日常的とは言え専門性をもって扱う理工系の研究者にとっては、(La)TeX 形式で一次元の文字列としてキーボードからいきなり入力するのが一番便利であろう。そのような専門性に依存せず、blog や wiki といった日常性において数式を扱う場面があるとすれば、中学から大学初年級までの生徒・学生や教育関連の人達によるものであろう。利用の対象をこのように限定すれば、扱う式の複雑さがある程度限定される一方、導入（あるいは導入に失敗するような事態）がハードルとしないようにすることや編集操作が直感的に理解できることが求められる。BrEdiMa はこれを基本理念として開発が進められてきたが、今回ブラウザでの数式書きの鉛筆&消しゴムとすることに当たり、汎用性と一般性が高いと思われる方法を用いた。その結果、導入のためのある程度の知識と手順が必要となってしまった。本稿では、これらに焦点を絞って説明を行った。数式入力は鉛筆&消しゴムという意味では、日本語入力システムになぞらえるべきかもしれない。数式のためのフロントエンドが日本語入力と同じような形態のものもあるが、数式が単なる文字列ではなく固有のデータ形式で表現され、表示にも固有のレンダリングが必要であることを考えると、すべてをフロントエンドの機能として押し込めることが適切かには疑問を感ずる。将来、数式入力の方法がどのような形に収束していくのかは未知数だが、BrEdiMa のように、固有のソフトウェアとして作り込まないのは一つの有力な方向であろう。

前説の最後で少し触れたとおり、現状の BrEdiMa では、数式の再編集は JSON というリスト形式で保存された場合にしか対応していない。そのため、blog や wiki の文中の数式の再編集ができるようにするには、blog や wiki に数式の扱いを拡張する専用のプラグインを用意するか、或は、多くが用いている LaTeX 形式の数式を BrEdiMa でも扱えるようにするなど、何らかの標準化と開発が必要である。BrEdiMa での対応法の検討と開発は早急に取り組むべき将来の課題である。

参考文献

- [1] Mathematical Markup Language(MathML) Version 2.0. W3C Recommendation 21 October 2003. <http://www.w3.org/TR/MathML2/>.
- [2] W3C Math Home. <http://www.w3.org/Math/>.
- [3] Y.Nakano and H.Murao: BrEdiMa: Yet Another Web-browser Tool for Mathematical Expressions.
<http://www.activemath.org/paul/MathUI06/proceedings/BrEdiMa.html>.
- [4] BrEdiMa の公式サイト
<http://bredima.sourceforge.jp>.
- [5] 研究室内の BrEdiMa のサイト
<http://w9-802.cs.uec.ac.jp/wiki.cgi?page=BrEdiMa>.
- [6] 中野泰人. 現代的 Web 技術の数式入力システムへの応用. 電気通信大学電気通信学研究科修士論文. 2008年3月.
- [7] Wei Su, P.S.Wang and Lian Li. Entering and Editing Mathematical Expressions on the Web. Mathematical User-Interfaces Workshop 2008.
<http://www.activemath.org/workshops/MathUI/08/proceedings/WeiWangLi-MathEdit.html>.
- [8] Firefox Add-ons. <http://addons.mozilla.org/ja/firefox/>.
- [9] Greasespot. <http://www.greasespot.net/>.
- [10] Dive Into Greasemonkey. <http://diveintogreasemonkey.org>.
- [11] Firebug - Web Development Evolved. <http://getfirebug.com/jp.html>.
- [12] Greasemonkey for IE Home Page. <http://www.gm4ie.com/>.
- [13] Trixie, Teaching an old browser new tricks. <http://www.bhelpuri.net/Trixie/>.
- [14] mimeTeX quickstart. <http://www.forkosh.com/mimetex.html>.
- [15] JSON の紹介. <http://www.json.org/json-ja.html>.

A BrEdiMa 導入ガイド – Greasemonkey 編

本文では、BrEdiMa がどのようなソフトであるかの説明を行った。ここでは、クライアント側でどのようにして使えるようにするかを、実例を示しながらわかり易く説明する。繰り返しになるが、サーバー側に設置するのではなく、インターネットの普通の利用者が個々に BrEdiMa を導入して使おうという場合である。

ブラウザとしては Mozilla Firefox を利用することとする (Greasemonkey によるブラウザの機能拡張を行うが、Internet Explorer 用の Greasemonkey での動作は未確認のため)。大まかな手順は以下のとおり。

- (1) Greasemonkey を Firefox にインストールする、
- (2) ユーザスクリプトである、BrEdiMa の JavaScript プログラムを用意する、
- (3) そのプログラムを Greasemonkey に取り込み、どの URI に対して拡張機能を動作させるかを設定する。

これらのインストール作業は一度だけやれば十分である。

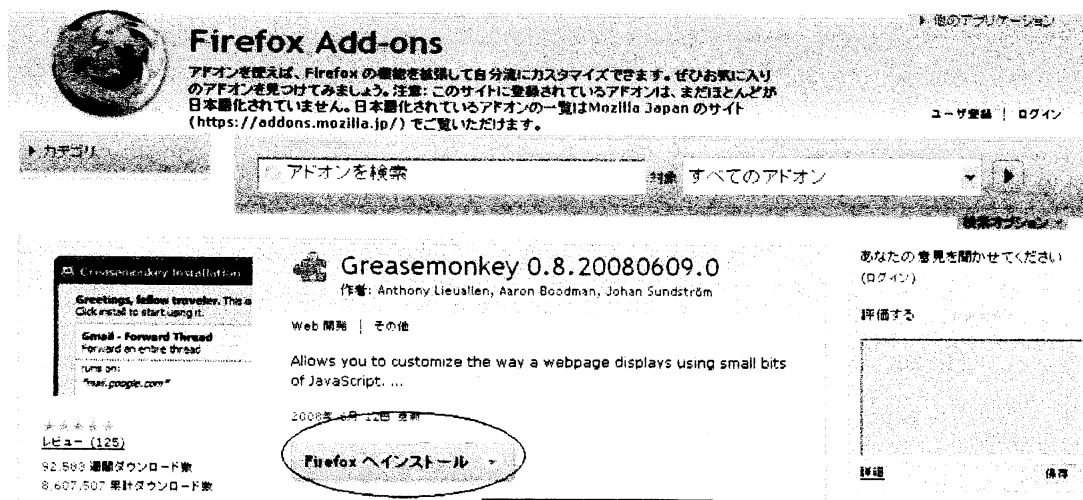
A.1 Firefox への Greasemonkey のインストール

まず、ブラウザの機能拡張を行うための環境を整える。⁶ 公式サイトダウンロードページにアクセスして直接インストールするのが最も簡単である。

- ブラウザ Firefox を起動し、公式のダウンロード・ページ

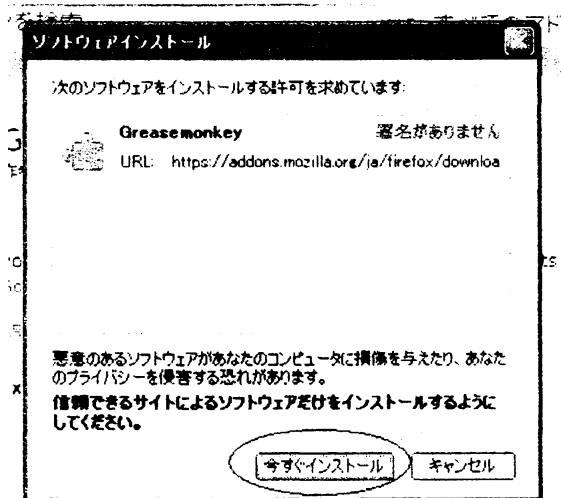
<https://addons.mozilla.org/ja/firefox/addon/748>

を開き、[インストール] ボタンをクリックする。



⁶ ブラウザ Firefox を起動し、そのウィンドウの右下に次ページに示す狼の顔が現れていれば (Greasemonkey は導入済みなので)、本節の作業は不要。

- ◇ 別の方法として、直接インストールするのでなく、ダウンロードしたファイルを用いて上と同等の作業を行うこともできる。
 - (i) 公式サイトダウンロードページ(上と同じ)からファイル `greasemonkey-xxx.xpi` (`xxx`の部分はバージョン、日付等)をダウンロードしておく、
 - (ii) ブラウザ Firefox を起動し、[ツール] → [アドオン] で現れたウィンドウにこの `.xpi` ファイルをドロップする。
- 下図のような確認の後にインストールが行われる。



指示に従い、ブラウザの「再起動」を行ってインストールが完了する。

- ブラウザの右下には猿の顔のアイコン(上右図)が表示されるようになり、無事インストールされたことが確認される。



(※) この猿の顔はクリックすることができ、Greasemonkeyの有効・無効が切り換えられる(灰色の猿が無効の状態)。

A.2 JavaScriptプログラムの用意

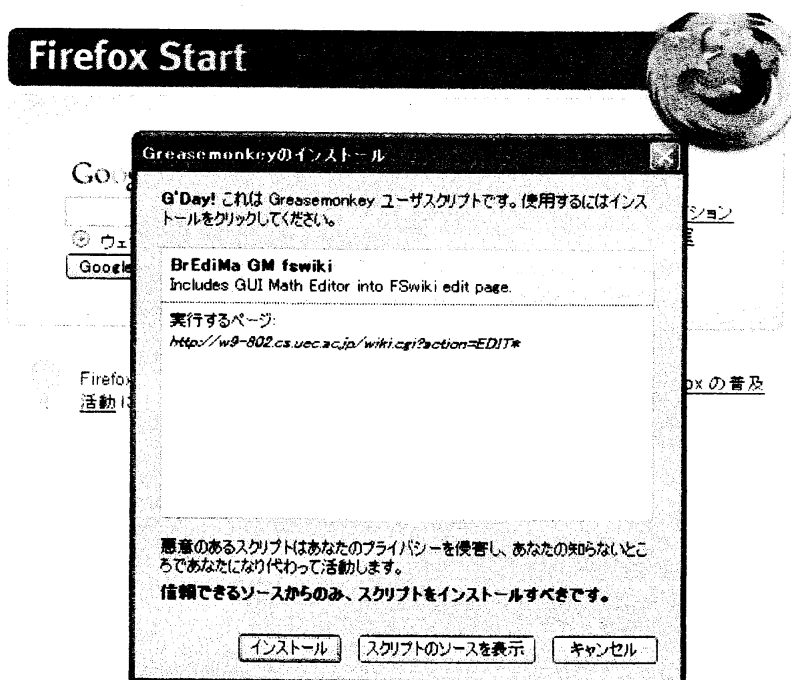
前節で導入した Greasemonkey は、それ自身でブラウザ本来の機能に変化をもたらすものではなく、ユーザが用意したプログラムを実行する環境を提供する。そのプログラムはユーザスクリプトと呼ばれ、JavaScriptで記述する。ここでは、そのようなプログラムのひとつとして BrEdiMa のユーザスクリプトを用意する。BrEdiMa はブラウザ上で数式を作成し Web ページへの入力とするためのソフトだが、どのような形で入力するかは対象とする Web ページに合わせる必要がある。即ち、blog や Wiki クローンのページで利用するには、適合する JavaScript プログラム(ファイル名は `—.user.js`)を準備して導入し、次節で示すようにページの種類に応じてどのプログラムを適用するのかを設定しておく必要がある。これまでに用意されている BrEdiMa の Greasemonkey 用ユーザスクリプトは次のとおり。

- はてなダイアリー：公式サイトである sourceforge のページ [4] の「ダウンロード」のページに在る `bredima_hatena.user.js`
- MediaWiki：筆者らの研究室の非公式ホームページ内の BrEdiMa のページ [5] に在る `bredima_mediawiki1.0.user.js`. MediaWiki は Wikipedia で用いられているが、この BrEdiMa は日本語用であり、他の言語用の MediaWiki での動作は未確認である
- FreeStyleWiki：上と同じページに在る `bredima_fsw1.0.user.js`

これら以外に対しては自前で用意することになるが、上のいずれかを入手して改造すれば良いだろう。改造は、blog や Wiki との連携をとる部分だけでそれほど難しくない。詳しくは、本文の § 3 参照。

A.3 Greasemonkey に取り込み設定する

- (3-a) 前節で用意した BrEdiMa のスクリプトのファイル (`---.user.js`) を、Greasemonkey が有効になっている状態⁷の Firefox のウィンドウにドラッグ&ドロップする。

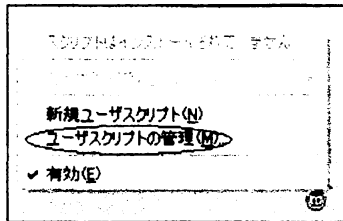


上左図のようなウィンドウが現れるので「インストール」をクリック。ウィンドウが現れずにブラウザ上にプログラムが表示された場合は、Greasemonkey が無効に

⁷ 猿の顔の画像が灰色でなく茶色の状態

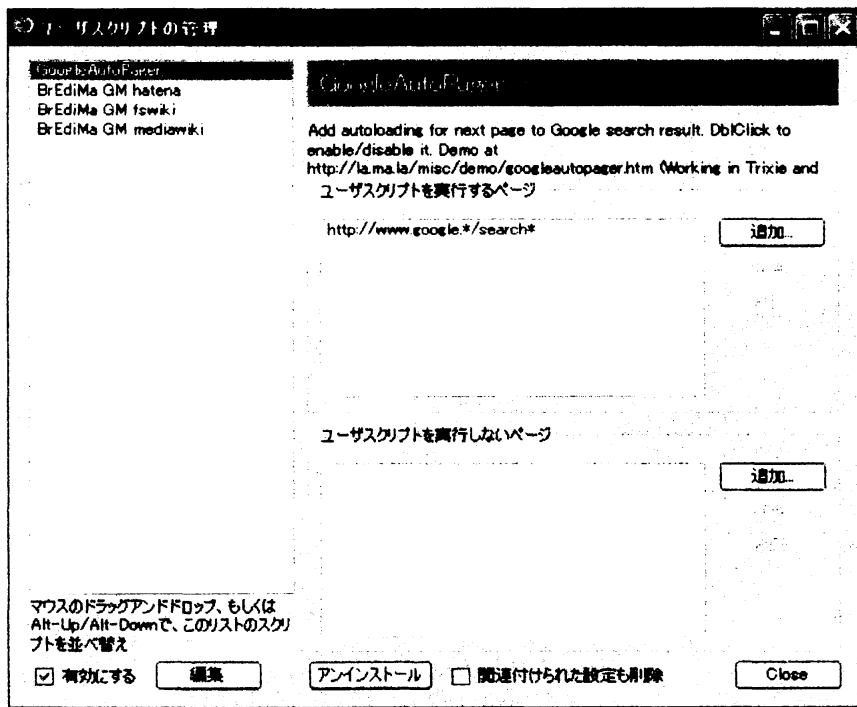
なっていることが原因なので、灰色の猿の顔をクリックして有効にした後（猿は茶色になっている）、再度ドラッグ&ドロップ。

- (3-b) 取り込み完了後、ブラウザの右下の猿の顔のアイコンを右クリックし「ユーザスクリプトの管理」を選択する。



新たなウィンドウが現れるので、スクリプト（のファイル）がインストールされたことを確認する。ウィンドウの内容は以下のとおり。

- 左側：取り込み済みのスクリプト（ファイル名）の一覧
- 右側：左側で選択したスクリプトを実行する/実行しない Web ページの URL の指定



URL の指定には、スクリプト中に記述されていれば、それが標準の値として設定されている。

スクリプト	設 定 (実行するページ)
はてなダイアリー用	<code>http://d.hatena.ne.jp/*/edit*</code>
Wikipedia/MediaWiki 用	<code>http://*/w/*&action=edit*</code>
FreeStyleWiki 用	<code>http://w9-802.cs.uec.ac.jp/wiki.cgi ?action=EDIT*</code>

- はてなダイアリー用のスクリプトでは、当然、スクリプトを実行するページの URL としてはてなダイアリーのサイトの編集ページに対応する文字列が設定されており、そのままではてなダイアリーで利用可能である。
- 現在用意してある Wiki 用のスクリプトでは適当な Wiki クローンを運用しているサイトに設定してあるので、各自で自分が利用したいサイトの URL を設定しなければならない。
 - 予め設定されているページの URL は不要となるので、削除するか編集して書き換える。
 - 自分が利用する Wiki ページの URL を記述するには、「追加」するか、予め設定されているページの URL を選んで「編集」する。
 - この記述には任意の文字列を表す総称文字「*」を使うことができる。Wiki の運用サイトは複数の Web ページで構成されるのが普通だが、その全てを指定するためにこの総称文字を用いる。
 - 更に動作を編集のページに限定するために、「action=EDIT」のように、Wiki の編集ページを表す固有の文字列を加えておくと良い。MediaWiki 用の設定は日本語の Wikipedia の編集ページに対応している。

これらは、標準の設定を真似るか編集して用いると良い。

スクリプト自体の修正は左下の「編集」ボタンから。