

# 次世代 Greasemonkey、Tsukikage System の紹介

中山心太

電気通信大学 電気通信学研究科 人間コミュニケーション学専攻  
182-8585 東京都調布市調布ヶ丘 1-5-1  
shinta@edu.hc.uec.ac.jp

**概要** Firefox の拡張機能に Greasemonkey というものがある。このソフトウェアはウェブページを閲覧している際に、任意の JavaScript を実行することができる。しかし JavaScript はセキュリティ上の都合からローカルファイルの読み書き、ネイティブコードの実行などができない。そこで本稿ではそれらを可能にした Tsukikage System を提案する。

## Introduction of next generation Greasemonkey, Tsukikage System.

Shinta Nakayama

The Department of Human Communication, The Graduate School of Electro-Communications,  
The University of Electro-Communications  
1-5-1, Chofugaoka, chofu-shi 182-8585, Japan  
shinta@edu.hc.uec.ac.jp

## 1 はじめに

Firefox の拡張機能のひとつに Greasemonkey<sup>1</sup>がある。このソフトウェアはウェブページを閲覧している際に任意の JavaScript を実行し、ウェブページを書き換えるものである。これを利用することにより、たとえば余計な広告を消したり、Amazon で閲覧中の商品を Yahoo!オークションで検索したりすることができる。しかしこのソフトウェアには限界がある。ウェブページを書き換えに JavaScript を利用しているため、自然言語処理やネイティブコードを利用した処理ができない。そこで本稿では、これらを改善した Tsukikage System の紹介を行う。

## 2 既存技術

### 2.1 Greasemonkey

Greasemonkey は Firefox の拡張機能で、閲覧中のウェブページにたいして任意の JavaScript を実行することができる。また、Firefox の特権モードで動作しているため、Firefox そのものを拡張したり、制限モードと違いドメインをまたいだ HTTP 通信ができるなど、高い拡張性がある。

### 2.2 他ブラウザの同様技術

ブラウザ	ソフトウェア名称
IE	Trixie <sup>2</sup>
Sleipnir <sup>3</sup>	SeaHorse <sup>4</sup>
Safari	GreaseKit <sup>5</sup>
Opera	UserJavaScript

表 1:Greasemonkey 相当のソフトウェア

他のブラウザでも、Greasemonkey と同様の働きをするソフトウェア、機能はいくつかある[表 1]。UserJavaScript は、Opera に標準で付いている機能である。それ以外はブラウザの拡張機能やプラグインとして実装されている。また、どのソフトも Greasemonkey との互換性を重視して作られており、Firefox 独自の JavaScript の拡張を利用したもので無い限り、Greasemonkey 用のスクリプトを利用することができる。

2 <http://www.bhelpuri.net/Trixie/Trixie.htm>

3 <http://www.fenrir.co.jp/sleipnir/>

4 <http://www.fenrir.co.jp/sleipnir/plugins/seahorse.html>

5 <http://8-p.info/greasekit/>

1 <https://addons.mozilla.org/en-US/firefox/addon/748>

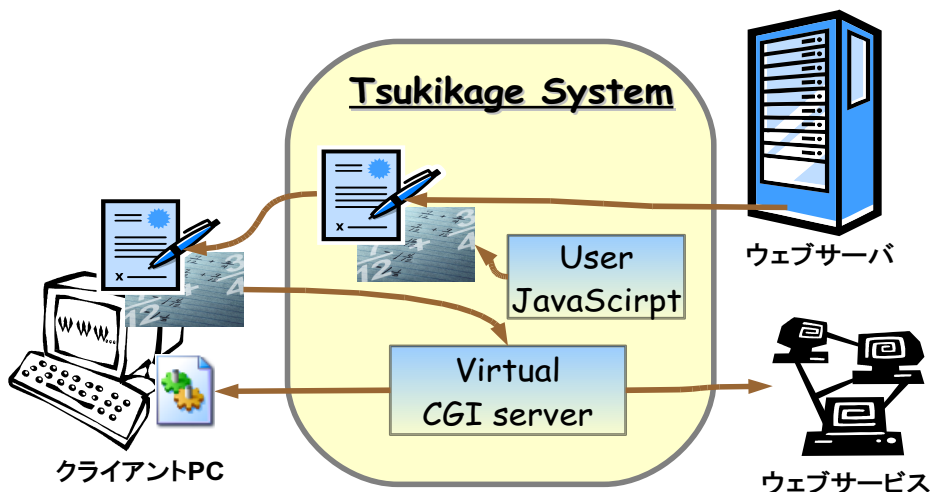


図 1:Tsukikage System の構造

### 2.3 proxomiton

proxomiton はローカル Proxy サーバとして働く。HTML をいったんバッファリングし、あらかじめ記述したルールに従って文字列を置換する。その後ブラウザに渡すことで、ウェブの書き換えを実現する。また、Proxy ベースであるため、通過する HTML 以外のデータも書き換えることができる。たとえばニコニコ動画ではコメントデータを HTTP で取得している。そのため、このデータを書き換えることで、Proxy レベルでのコメントのフィルタリングを行うことができる。Proxy ベースでの書き換えはブラウザを選ばないという特徴があり、非常に有用である。

## 3 従来手法の問題点

### 3.1 Greasemonkey 方式

JavaScript は動作速度に問題があり、書き換えのための複雑なアルゴリズム、たとえば形態素解析などを利用することができない。また、書き換えのためのツールは HTML の木構造を利用した DOM (Document Object Model) 操作と正規表現しか提供されていない。そのため、テンプレート生成のサイトなど、十分に構造化されたサイトでは有効に書き換えることができるが、構造化されていないサイトでは、文章の中身を書き換えることは非常に難しい。

### 3.2 proxomiton 方式

複雑なアルゴリズムを利用してデータの書き換えを行うと、時間がかかりレイテンシーが低下する。そのため、proxomiton では複雑なアルゴリズムを利用することができず、ルールベースでの書き換えしか行うことができない。

## 4 Tsukikage System の提案

Tsukikage System は Greasemonkey とほぼ同等の機能を持ちながら、JavaScript から Python スクリプト、Python スクリプトからネイティブコードの呼び出しを可能にする。これにより、JavaScript の欠点であった実行速度、自然言語処理の問題が解決する。

### 4.1 Tsukikage System の実装

沖ソフトウェア株式会社<sup>6</sup>で MIT ライセンスで公開されている Tiny HTTP Proxy<sup>7</sup>を元にして実装を行った。Tiny HTTP Proxy は Python で記述されているため、その上でプラグイン形式で Python のコードを実行することは非常に容易である。

【図 1】は Tsukikage System のデータフローである。それぞれの実装は次項で説明する。

#### 4.1.1 Proxy による JavaScript の付加

Tsukikage System は proxomiton と同様にローカル Proxy サーバとして動作する。しかしルールベースでの書き換えではなく、HTML のフッターに JavaScript を付加することで、HTML の書き換えを実現する。

図 1 中の User JavaScript が付加するための JavaScript である。この JavaScript は Greasemonkey のメタ記法をそのまま利用することができる。そのため、どのページに何のスクリプトを利用するかというコントロールは非常に簡単である。またメタ記法は一部拡張しており、prototype.js といったライブラリの利用、IE と Firefox で別のスクリプトを使うなど、ブラウザごとの付与スクリプトのコントロールを可能にしている。

6 <http://www.okisoft.co.jp/>

7 <http://www.okisoft.co.jp/esc/python/proxy/>

#### 4.1.2 Proxy とブラウザの通信

JavaScript の httpXMLRequest はセキュリティ上の制限から、ドメインをまたいだ通信は行うことができない。しかし同一ドメインで架空のファイルならば命令を実行することができる(行うと 404 エラーが返ってくる)。そのため、Proxy が架空アドレスへの通信をフックすることによって、任意の情報を返答することができる。これにより AJAX と同じ原理でウェブページを書き換えることができる。

#### 4.1.3 Python コードの利用

架空アドレスへの通信をフックした Proxy は、その内部に持った仮想 CGI サーバ上の Python スクリプトを実行する。図 1 中の Virtual CGI Seaver がこれにあたる。これにより JavaScript から Python スクリプトを実行したように振舞うことができる。

また、Tsukikage System はユーザー権限で動作しているため、その中の Python スクリプトもユーザー権限相当のことができる。たとえば他のウェブサーバにアクセスして情報を収集したり、他のアプリケーションへメッセージを送信したりすることができる。さらにネイティブコードで記述された DLL を呼び出すことによって、ネイティブコードでなければできないような速度、ハードウェアへのアクセスを実現する。

### 4.2 Proxy 実装による恩恵

JavaScript を拡張して他のスクリプト言語やネイティブコードを利用できるようにするのは、Greasemonkey 系のソフトウェアを拡張することでも容易に実現することができる。しかし Proxy で実装することにより様々な恩恵を得る事ができるため、今回は Proxy で実装を行った。

#### 4.2.1 様々なブラウザへの対応

従来の Greasemonkey 系のソフトウェアの場合、JavaScript のライブラリを利用するにはある程度複雑な手続きを踏まなければならなかった。しかし UserJavaScript のメタ領域を拡張し、ライブラリのインポートを容易にし、ブラウザの差異をライブラリで吸収可能にした。また同様に、メタ領域にブラウザ判定を設けることにより、ブラウザごとに別のスクリプトを実行可能にした。これにより、様々なブラウザから利用可能になった。

#### 4.2.2 非 PC ブラウザへの対応

表 1 は標準的なブラウザ用のソフトウェアである。PSP や Wii などのブラウザではユーザーが開発したプラグインやスクリプトを導入することができない。しかし Proxy の設定はできるため、これを利用することによって、Greasemonkey 相当の機能を得る事ができる。

#### 4.2.3 グループウェアとしての利用

Proxy なので、複数のユーザーから同時に接続することが可能である。そのため、複数のユーザーの行動履歴を一元管理し、それをもとにウェブを書き換えることができる。これにより、ウェブサイトすべてをグループウェア化することが可能になった。

## 5 サンプルアプリケーション(デモ中)

下記ソフトウェアは発表会場でデモしています。詳細はポスターまで。

### 5.1 株価表示

現在のウェブニュースサイトでは、事件や事故などのニュースは株価材料のページに乗らず、一般ニュースのカテゴリに記載される。株価材料のページ以外では株価情報サイトへのリンクが張られていることはあまり無い。しかし事件や事故が起こったら真っ先に株価を調べたいというのが投資家の心情である。そこで、ウェブページ中に上場企業の名前が登場したら、その企業の株価情報のページへのリンクを表示するというアプリケーションを開発した。

ウェブページを MeCab<sup>8</sup>を利用して形態素解析し、ハッシュテーブルに確保した企業名リストと一致するか調べる。形態素解析はおおよそ線形時間であり、ハッシュテーブル参照は定数時間であるため、処理全体としては線形時間で探索することができる。

形態素解析を行わずに find を利用して企業名が含まれていないか判定を行った場合、計算量はおおよそ文章量\*企業数という形になる。速度を調べたところ上場企業程度(約 3700 社)では形態素解析を利用した場合とほとんど差は無かった。しかしこれが Wikipedia のインデックス数のような規模(40 万強)になると、形態素解析を利用したほうが圧倒的に速くなる。また find では文章中に「フジテレビ」が存在した場合、「フジテレビ」と「フジ」という両方の企業が検出されてしまう。形態素解析を利用すると「フジテレビ」という名詞として検出されるため、このような現象は起こらない。

### 5.2 付箋ソフトウェア

Tsukikage System はグループウェアとして利用することが可能である。そこでウェブページに任意に付箋的なコメントを付けれるニコニコブックマーク<sup>9</sup>を参考にし、同様の機能の実装を行った。

ウェブを閲覧している際に任意の場所でダブルクリックをすることで、コメント入力用のフォームが出現する。これにコメントを書き込むことで Tsukikage

8 <http://mecab.sourceforge.net/>

9 <http://www.nicob.jp/> (サービス終了)

System 上の仮想 CGI サーバにデータが送信され、CGI サーバ上のメモリに保存される。またブラウザ側で定期的にポーリングし、書き込まれたコメント情報を取得することで画面に表示させることができる。

### 5.3 マウスカーソルの共有

2006 年度上期未踏コースに採択された Antwave<sup>10</sup>を参考にし、同じページを見ている人のマウスカーソルを共有するソフトウェアを開発した。原理としては付箋ソフトウェアとほぼ同等で、Proxy に定期的にマウス座標を JavaScript で送信し、他のユーザーのマウスカーソル座標を受信する。これにより他のユーザーのマウスカーソルを表示させることができる。

### 5.4 フィッシング対策ソフトウェア

中山が開発したアルゴリズム[1]をもとにフィッシング対策ソフトウェアを開発した。人はフィッシングサイトの疑いのあるページに出くわしたら、そのページのタイトル(騙っている企業名)を Google で検索し、正規サイトを見つけ出し、URL が異なっていないかどうか調べる。この手法は Google を一種のホワイトリストとみなし、九割以上の確率で正確な判断を下し、従来のフィッシングサイト検出ソフトとほぼ同等の性能を示す。

ページタイトルの取得はタグから拾えば簡単にできるように思えるが、場合によっては「Welcome!」や「ようこそ」となっており、タグから企業名を判断することは非常に難しい。そこで自然言語処理を行い、ページの本当のタイトルを見つけ出す必要がある。そこで形態素解析と TF-IDF 法を組み合わせこれを実装した。

## 6 今後の課題

### 6.1 セキュリティ

HTML のフッターに JavaScript を付与しているため、ブラウザ側から見ると、Tsukikage System が付与したコードなのか、もとからあるコードなのか区別することができない。そのため、HTML 中に Tsukikage System を悪用するコードを埋め込まれる可能性がある。

### 6.2 ルールベース書き換え

Proxy ベースで実装しているため軽い処理であれば proxomiton と同様に Proxy 通過時にも書き換えを行うことができる。今回は Proxy と通信を行って書き換えを行ったが、事前にページ加工をすることで、ブラウザの JavaScript による負荷を下げるなどがで

きるため、こちらの実装も検討している。

## 7 まとめ

Greasemonkey のウェブページ書き換え機能をさらに拡張した Tsukikage System の提案した。

Tsukikage System はウェブページ書き換え時に JavaScript から Python のコード、Python のコードからネイティブコードを利用できる。これにより、ローカルファイル IO やハードウェア制御ができるようになった。

ハードウェアコントロールができるため、たとえば iPod とウェブの連動などができる。大島聡史氏によって Tsukikage System 上のアプリケーションとして開発された NICOX というものがある。これは、ニコニコ動画<sup>11</sup>用のダウンロードツールであり、ウェブブラウザ上から PSP に動画を保存できるというものである。従来のブラウザへ組み込む形のダウンロードツールはニコニコ動画で配信されているフラッシュビデオ形式(FLV)でのダウンロードはできていたが、MPEG4 への変換はできなかった。またニコニコ Podder<sup>12</sup>のような専用のダウンロードツールでは、ダウンロードと MPEG4 への変換はできるが、ブラウザに組み込んで行うことができなかった。しかし NICOX はブラウザに組み込むことができ、ダウンロードと同時に FFmpeg<sup>13</sup>を利用して MPEG4 への変換を行うことができる。

従来のウェブはセキュリティ上の都合から、ローカル環境と切り離されており、そのためウェブ上で複雑なことを行うことができなかった。そのため、すべての操作をウェブ上で完結させるためにウェブ OS という概念が生まれた。しかし Tsukikage System は NICOX の例のように、ローカルとウェブの融合を可能にした。これにより、新しいアプリケーションの可能性を提示することができた。

## 謝辞

本研究は情報処理推進機構(IPA)の未踏ソフトウェア創造事業において、「2006 年度下期未踏コース」の支援を受けている。

## 参考文献

[1]中山心太,吉浦裕,“模倣コンテンツの特性に基づくフィッシング検知方式”,第 38 回 CSEC 研究会

11 <http://www.nicovideo.jp/>

12 <http://www.rocaz.net/>

13 <http://ffmpeg.mplayerhq.hu/>

10 <http://www.ipa.go.jp/jinzai/esp/2006youth/gaiyou/1-10.html>