

# Web における front-end——『stir』の開発 —Web ページ勝手組み換えツールの開発—

## 1. 背景

近年、ユーザが主導的にコンテンツを作成するタイプの Web サービスが人気である。例として、会員制の日記共有サービスである『mixi』や動画共有サービスの『YouTube』、不特定多数の編集者による百科事典作成プロジェクト『Wikipedia』などが挙げられる。これらのサービスにおいては、ユーザが日記や動画・記事などのコンテンツを作成し、コンテンツ提供者側はユーザがサービス上で活発に交流を行えるように、プロフィールや掲示板等の機能を提供する役割分担がされている。このように、実際に情報を生成しているのがユーザであることがこれまでの Web サービスと異なり、自分の好みに応じたコンテンツを作成できることが多くのユーザからの支持を得た。

しかしユーザがこれらのサービスを使用していると、しばらくしてその不自由さに気づくことになる。自分が作成する日記や記事といったコンテンツの中で、より表現力の高い CSS(Cascading Style Sheet)や JavaScript の使用がコンテンツ提供者側から許可されていなかったり、一度に送信できるファイル数が限られていたりする。結局のところ、これらの新しいサービスの中でも、ユーザはコンテンツ提供側が予め設定した枠組みの中で自由を謳歌しているに過ぎない。

これらのサービスの台頭によって、これからの Web コンテンツを主導的に生成していくのは、Web サービスの提供側ではなく個々のユーザであることが示唆された。この考え方をさらに発展させ、提供された Web コンテンツをユーザの手によって、個々人の事情に即した形に組み替えられる仕組みを提案することが本プロジェクトの趣旨である。

このようなことを実現させる仕組みとしては、すでに『Greasemonkey』という、Web ブラウザの Firefox に追加機能として動作するものは存在する。しかしこのアプリケーションは、そもそも Firefox というブラウザ上でしか動作しないこと、コンピュータ間での設定の共有ができないこと、複雑な設定が必要で初心者には敷居が高いことなどの問題があった。これらの問題を本プロジェクトではシステム全体を Web アプリケーションとして作成することで、Firefox だけでなく幅広いブラウザで動作できるようにしてブラウザ依存の問題を解決し、サーバ上でユーザ情報を保持することで設定共有の問題を解決し、基本的な追加機能についてはシステム側から提供することで、誰でも手軽に使用することが可能なシステムを目指す。

## 2. 目的

Web 上に存在する既存のコンテンツを横断的に利用するためのフレームワークを開発する。このフレームワークを使用することで、ユーザはこれまでコンテンツ提供側が独自に決めていたサービスの形や適用範囲を、ユーザ個々人の事情に即して組み替えて使用すること(例:新しい機能の追加・スタイルの変更)が可能になり、さらには、コンテンツ提供側の制限を越えて使用すること(例:ブックマーク数の制限解除・HTML タグの使用)も可能になる。このフレームワークを Web アプリケーションとして作成することで、インストールの手間がなく、誰でも手軽に使用することが可能なシステムとして構築する。

### 3. 開発の内容

#### 3.1 仮想ブラウザ機能

JavaScript の http 通信機能を用いて、指定された URI に存在する Web コンテンツを取得する機能が実装されている。Web サーバから取得した情報に対して動的にリンク補完やスタイル設定の適用・スクリプトの実行等を行っている。このように本システムはユーザと Web コンテンツとの間で薄いフィルタのように動作し、ユーザ側が特殊な操作を必要としないので、ユーザは通常のブラウジングと同じ感覚で本システムを使用することが可能である。

#### 3.2 サーバとの連携機能

本システムは仮想ブラウザの設定を Web 上に保存する。これはユーザがシステムを使用している間に行われ、ユーザが設定の保存について意識することはない。

#### 3.3 コンテンツの組み換え機能

本システムの特徴の一つとして、追加機能を適用することで、ユーザが個々人が既存の Web サービスを組み換えることが可能となる機能を実現する。これは Firefox の Greasemonkey に触発されて追加された機能であるが、本システムでは Web アプリケーションとして実装することで、元の Greasemonkey では実現不可能だった、追加機能を異なるコンピュータ間で共有することが実現できた。

#### 3.4 デモ・開発用 Web サイト

本システムのデモを体験可能な Web サイトを構築した。アンケートや掲示板により、システムや機能の開発者がユーザと交流を図れるようになっている。

#### 3.5 使用の手引き

本システムを始めて利用するユーザのために、導入の手引きとなる文書を作成し、デモ用の Web サイトで公開した。

### 4. 従来技術(または機能)との相違

本システムの主たる特徴は、ユーザ側が既存の Web コンテンツを個々人の事情に即して、組み替えたり付け足したり、あるいは削ったりできるところにある。既存のコンテンツを丸々利用するので、本システムのために新規コンテンツを開発する必要がなく、現在世の中に無数に存在する Web コンテンツを本システム上で利用することが可能なのは、本システム最大の強みだ。さらに、Greasemonkey と同じ API(Application Program Interface)を本システムに実装し、Greasemonkey 用に開発された機能がそのまま本システムで動作可能とした。

## GreasemonkeyAPIの対応状況

- GM\_log
- GM\_setValue
- GM\_getValue
- GM\_xmlHttpRequest
- × GM\_addStyle
- × GM\_openInTab
- × GM\_registMenuCommand

掲載資料: <http://files.getdropbox.com/u/1410212/stir7.ppt>

図 1 GreasemonkeyAPI の対応状況

本システムはWebアプリケーションとして作成されたことで、異なるOS・ブラウザ間の差異の吸収に留まらず、別のコンピュータにおいてユーザが同一の設定で本システムを通したWebブラウジングが可能なシステムとなった。これによって、ユーザが職場や学校と自宅で別々のコンピュータを使用している場合でも、同一の設定における本システムを提供することが可能になった。

Webコンテンツをユーザが組み替える機能を提供するという点で本システムと類似した、GreasemonkeyとiGoogleという既存のアプリケーションとの対比を以下に示す。

サービスの比較			
	Greasemonkey	iGoogle	stir
環境依存性	×	◎	◎
初心者の導入の容易さ	△	△	◎
設定の共有	×	◎	◎
追加機能の組み合わせ	◎	×	◎

◎…得意      △…不得意  
○…可能      ×…不可能

図 2 本システムと類似アプリケーションとの比較

GreasemonkeyはFirefoxブラウザでしか動作せず、また起動ごとに追加機能のアップデートチェックが必要なことなど、環境依存性が非常に高い。この点についてiGoogleはシステムをWebアプリケーションとして作成することで、この問題を解決しており、本システムでも同様の手段をとった。初心者の導入については、Greasemonkeyは追加機能を自分で探してくる必要があるため敷居が高く、iGoogleでもその手間はほとんど消えていないが、本シ

システムでは有用な追加機能はシステム側が提案するため、初心者導入の敷居が他システムに比べて格段に低くなっている。設定の共有については、Greasemonkey は同一コンピュータの同一ブラウザ内でしか動作できないので不可。iGoogle と本システムは Web 上に設定を保存するので得意となっている。追加機能の組み合わせとは、一つの Web コンテンツに対して複数の追加機能がある場合に、それらを同時に適用することができるかということである。これは Greasemonkey の得意分野であり、それを継承した本システムも同様である。しかし iGoogle は元々そのような使われ方を想定しておらず追加機能の組み合わせは難しい。

異なるドメインに存在する Web コンテンツの横断的な利用という観点では、近年 OpenID という複数の Web コンテンツ間でユーザの情報を共有するサービスが出現した。しかしこのサービスはコンテンツ作成者側に向けてのサービスであり、Web コンテンツ側が対応しないとユーザはその恩恵を受けられない。本システムはユーザ主導で Web コンテンツを横断的に利用するために、そのような仕組みに対応していない Web コンテンツであっても利用できるという強みがある。

## 5. 期待される効果

本プロジェクトで作成された『systir』により、これまで Web サービスを享受するだけの存在であったユーザ側が、逆に Web サービスを組み替えたり、さらにはその組み替えたサービスを公開することができるようになった。これにより、今後 Web サービスを特定の用途に向けてカスタマイズして公開する人が出てきたり、あるいは組み込む“機能”の部分だけを専属に開発して利益をあげるようなことができるようになることが期待される。

## 6. 普及(または活用)の見通し

今後は基本となる追加機能を増やし、「いつもの Gmail や mixi にちょっと機能が付け足されるサイト」として周知させる。過去の Greasemonkey の例では、この段階に 1~2 年かかると見られるが、導入が簡単な本システムではさらに短い期間でのユーザへの浸透が望める。その後の段階で追加機能をユーザも作成できることを説明し、自分で追加機能を作成できる上級ユーザを会得しようと考えている。そのような上級ユーザ用のユーザ・コミュニティの場として掲示板等の機能が本システムの中で用意されている。

このような成長戦略をとることで、Greasemonkey の現在の平均ユーザ数である約 300 万人を超えるユーザ数の会得を、2 年で実現したいと考えている。

## 7. 開発者名(所属)

米倉健太(早稲田大学)

<http://www.systir.net/>