

## ブラウザ操作の自動再生システムの構築

椿 晃一郎<sup>†</sup>疋田輝雄<sup>‡</sup>明治大学  
理工学研究科<sup>†</sup>明治大学  
理工学研究科<sup>‡</sup>

## 1. はじめに

今日のウェブでは Ajax や HTML5 などの新しい技術が次々と登場し、ウェブブラウザの重要性が増してきている。そのような中、デジタルデバイスや、ウェブに依存する習慣的な反復操作 (WEB-EDI など) による人間側の負担増大が問題となってきた。

本研究では、こうした問題の一つの解決策として、ブラウザの操作を記録し、自動で再生できるシステムをウェブアプリケーションとして構築する。このシステムに必要なものはウェブブラウザのみで、プラグイン等は不要である。ユーザはウェブ上の本システムを利用して外部サイトにおける自らの操作を記録し、再生することができる。記録は非同期で行なわれ、再生ではマウスの動きなども忠実に再現される。外部サイトへのアクセスはシステム内のキャッシュシステムを通して行なう。

## 2. システムの概要

本システムは前述の通りウェブ上で利用する。システム上には専用のブラウジング領域が設けられており、利用者はその領域を使用して操作の記録、再生を行なう。本システムで記録、再生できる操作は主に以下ようになる。

- マウス、スクロールバーの動き
- ページ遷移
- 戻る、進むボタン
- フォームの送信

図 1 から分かる通り、システムは大きく二つのフレームに別れており、左側が操作フレーム、右側がブラウジング領域となっている。操作フレームには記録データが集められた再生リストとシステムの機能を使用するためのタブパネルが設置されている。各タブの内容と再生リストを図 2 に示す。

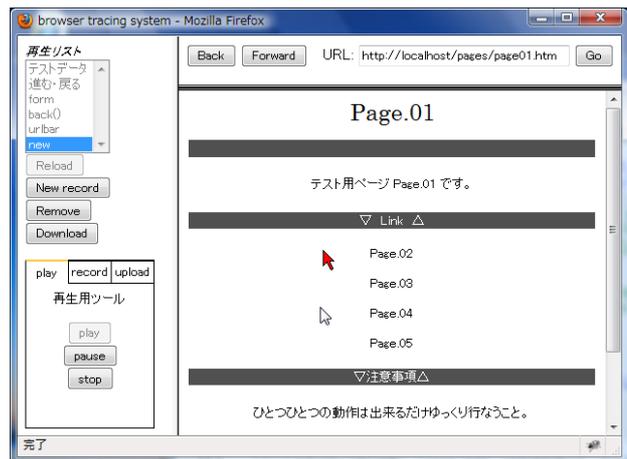


図 1 システムの GUI 全体像

## 2.1 ブラウザ操作の記録

ブラウザ操作の記録は主に record タブで行なう。記録時に必要なことは以下の 2 つである。

- 記録するデータの選択
- record タブの record ボタンのクリック

記録するデータは再生リストから選択するか、新規に生成することで指定できる。記録が開始すれば通常通りウェブサイトを閲覧するだけで、その間に行なわれた操作は非同期に記録される。ただし、戻るボタン、進むボタン、アドレスバーなどの機能はシステム内に特別に設けられたものを使用する。記録を終了するには、record タブの stop ボタンをクリックする。

## 2.2 ブラウザ操作の再生

ブラウザ操作の再生は主に play タブで行なう。再生時には再生リストから再生したいデータを選択し、play ボタンをクリックする。再生が開始されると、赤いマウスポインタが再生用として出現し、これが記録時のマウスの動きを再現してくれる。再生を一旦止めたいときは pause ボタンをクリックすることで一時停止することができる。再生を再開するには pause ボタンと同じ場所に出現する再開用のボタンをクリックすればよい。また、データの最後までトレースし終わると再生は自動的に終了するが、途中で再生

「Automatic Tracing System for Web Browsing」

<sup>†</sup> K. Tsubaki, Meiji University

<sup>‡</sup> T. Hikita, Meiji University

を中断したい場合は play タブの stop ボタンをクリックすることで終了することができる。



図2 各タブの内容と再生リスト

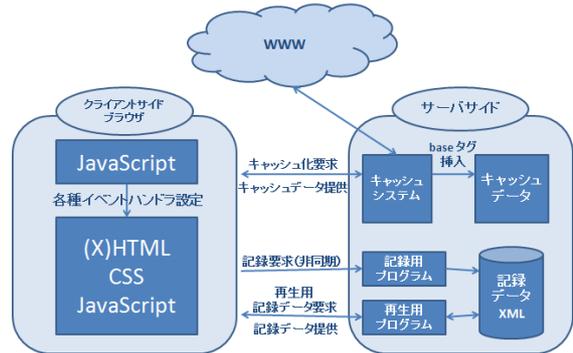


図3 システムの実装概略

### 2.3 その他の機能

本システムには記録、再生の他にも、記録データのアップロード、ダウンロード、削除などのような機能が備わっている。

### 3. システムの実装

本システムはウェブアプリケーションとして構築しているため、その開発には JavaScript が重要な役割を果たしている。しかし JavaScript で扱えるページにはセキュリティ上の制約が存在し、同一出身でなければ扱うことができない。

そこでこのシステムでは外部のサイトにアクセスする場合、一旦システム内にキャッシュを生成し、同一ドメインとすることで JavaScript によるデータへのアクセスを可能にしている。システムの実装概略を図3に示す。

#### 3.1 キャッシュシステム

キャッシュシステムは謂わばプロキシのような役割を果たす。キャッシュページにはシステムと同一ドメインでかつ元の外部ドメインへの参照を行なえるように base タグの埋め込みが行なわれる。また、フォームなどで用いられる GET 送信や POST 送信はサーバ側のプログラムによって返ってくるコンテンツが様々なものとなるので、専用のキャッシュプログラムを呼び出す。

#### 3.2 記録・再生用イベントハンドラの設定

ブラウザ操作の記録と再生は閲覧するページに設定された各種イベントハンドラによって行なわれる。システムはいくつかのフレームによって分割されているが、各種イベントハンドラはこのフレーム群の最上位フレームにあらかじめ読み込まれている JavaScript によって設定される。

再生時には play ボタンがクリックされたときに再生用イベントハンドラが指定された記録データを非同期に取得しに行き、その後データの

解析とともに再生処理を開始する。再生時もキャッシュシステムを通して外部サイトへアクセスするため、本システムでの再生は実際に外部サイトとデータのやり取りが行なわれる。

#### 3.3 記録フォーマット

記録データは汎用的な XML 形式で保存されている。以下に記録フォーマットの概略を示す。

```
<tracedata username="webmaster" number="2" ex ~>
  <data type="userInfo" ~ />
  <data type="coordinates">3,378;s0,102;432,390;s0,204;</data>
  <data type="link">{ "number": 0, "href": "(URI)" }</data>
  <data type="form">{"number":0, "text":[{"number":0,
    "name":"na", "id":"na", "value":"椿"}, ~}</data>
</tracedata>
```

システムにアップロードされる記録データは、このフォーマットに準拠した XML ファイルでなければならない。

### 4. おわりに

ブラウザ操作を自動化するソフトウェアはいくつか存在する。しかしどれもインストールが必要な単独ソフトウェアやブラウザの拡張機能を使用するものであり、特定の環境に依存してしまう。これに対して本システムではブラウザのみを使用するウェブアプリケーションであり、利用者に手間をかけさせない。また、初心者への操作支援という目的から、他の自動化ソフトウェアと違ってマウスなどの動きを可視化し、忠実に再現することを重視している。

このようなウェブを中心としたアプリケーション形態はクラウドコンピューティングという概念の根幹であり、重要な利点である。Google の Chrome OS などを見ても、今後この形態が主流の一つとなっていくことは間違いないだろう。

#### 参考文献

[1] 『JavaScript 第5版』 オライリージャパン  
 [2] 『動的ウェブページのブラウザ共有』 中村, 足田 DICOMO 2010.