

# クライアント間の動作の共有と可視化による Web アプリケーション学習支援

石井 正浩<sup>†</sup> 疋田 輝雄<sup>†</sup>

明治大学理工学研究科<sup>†</sup>

## 1. はじめに

今日ウェブ技術 Ajax は非常に大きな意味を持つようになった。しかし、Ajax には学習が困難であるという問題がある。Ajax はサーバとの通信や DOM の操作を主とするため実際に何をやりとりしているかクライアントからは見えにくいためである。また、Ajax を講義で解説する際には、一般的な紙媒体の教材ではアプリケーションの動作がわかりにくいいため、教員と学習者の実際の操作を共有できることが望ましい。しかし、文献[1][2]の方式では同一出成元ポリシーにより、Ajax 通信の共有が技術的に困難である。

本研究では、上述の問題を解決するために、動作の共有と可視化を可能にする Web アプリケーション学習支援システムを提案する。

## 2. 本システムの目的

本研究では、学習に必要な情報を可視化し、できるだけわかりやすい Web アプリケーション学習支援ツールを実現する。すべてを可視化するだけでは最近のブラウザに付属しているデバッグツールと大差がなく、また学習には向いていないためである。

また、クライアントのブラウザ間の共有化をすることにより、教員と学習者の間で同じ画面を見ながら効率的に学習を行うことを可能にする。

## 3. 動作例と操作

本システムの動作例とユーザインタフェースを図 1 に示す。①は「コントロール領域」で、ブラウジングの際に必要な URL を入力するバーがある。②は「アプリケーション表示領域」である。③は「情報可視化領域」で、ここにイベントなどの必要な情報を可視化し表示する。

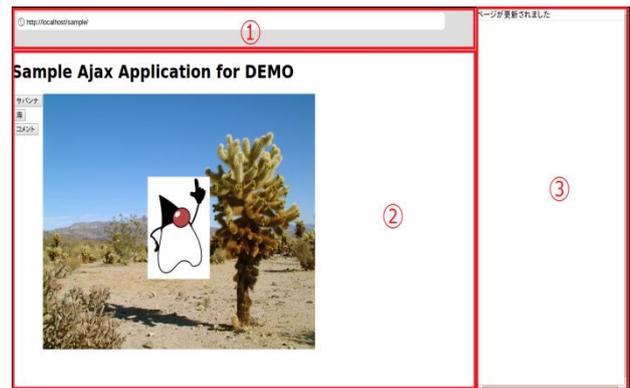


図 1 ユーザインタフェースとサンプルアプリケーション

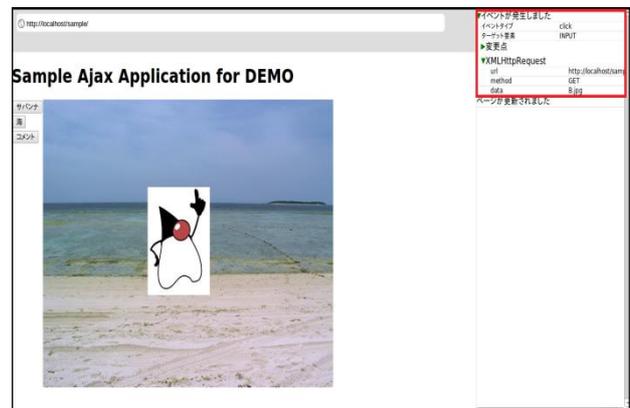


図 2 動作の様子

今回サンプルとして Ajax を用いる Web アプリケーションは、左側にボタンが 3 つあり、それぞれを押すことで背景画像が変わる。実際にはボタンを押すと XMLHttpRequest が発生し、サーバから変更する背景画像の URL を取得することで、その URL に基づいてキャラクターの背景画像が変化するというものである。図 1 のアプリケーション表示領域に表示されているのは、サンプルアプリケーションを本システムに読み込ませたものである。2 番目のボタンを押した後の様子を図 2 に示す。ボタンを押すことで背景画像が変わる。

図 1 と図 2 の間では、まず 2 番目のボタンに

Web Application for Learning Support by Sharing Behaviors  
Among Clients and Visualization.

<sup>†</sup>Masahiro Ishii, Teruo Hikita

School of Science and Technology, Meiji University.

対するクリックイベントが発生し、XMLHttpRequest で画像の URL を取得してページの DOM を書き換えるということが行われている。これらのイベントなどの中で実行の様子を可視化すべき情報が図 2 の赤枠内に表示される。一番上には、起こった JavaScript イベントについての詳細を表示する。この場合はクリックイベントであることとクリックされた html 要素である。次はその中で起こった XMLHttpRequest について表示する。今回の場合は通信先として画像の URL が入ったテキストファイルへの URL、通信メソッドとして GET が用いられたこと、受信したデータの中身を表示している。

## 4. システムについて

### 4.1. システム概要

本システムの概略図を図 3 に示す。本システムでは、サーバサイドでレンダリングを行い、その見た目の情報をクライアントに転送する。またサーバ上で起こることのうち、表 1 で示すような情報をクライアントに通知し、それを表示する。

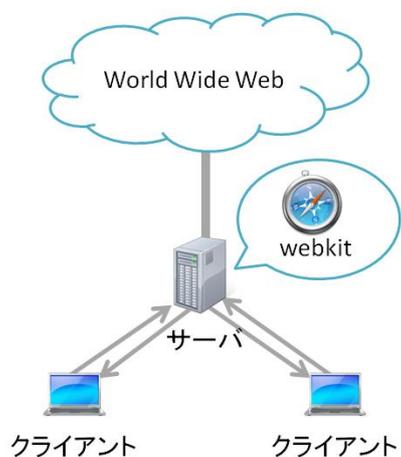


図 3 システム概略

表 1 システムで可視化する情報

可視化する情報の種類	可視化する情報
JavaScript イベント	イベントが起こされた HTML 要素 イベントの種類
DOM の変更	HTML 要素 HTML 属性の値 CSS プロパティの値
XMLHttpRequest	通信先の URL 通信メソッド(GET, POST など) 受信したデータ
ページ遷移	遷移先の URL

### 4.2. 可視化の方法と実装

可視化は JavaScript において、ブラウザが実装する JavaScript オブジェクトをラップして機能を追加することで実現する。あらゆる JavaScript オブジェクトに対しても、同一手法で実装することで情報を可視化することが可能である。

本論文で実装したシステムでは、Node オブジェクトや XMLHttpRequest オブジェクトをラップすることで、表 1 の情報を取り出せるようにしている。

### 4.3. 動作共有の方法と実装

動作共有は次の手順によって実現する。まずクライアント上で起こったイベントをフックし、その情報をサーバに送信する。次にサーバは受け取った情報を元にサーバ上のブラウザで同じイベントを実行して、その結果 (HTML+CSS) をクライアントに送信する。最後にクライアントは受け取った HTML, CSS を画面に表示することで動作の共有を実現する。この方式の利点はイベントさえフックできればすべてのブラウザ上での動作は共有可能になり、クライアント間における見た目の不整合も発生しない点である。

クライアントの実装としては、サーバに必要な情報を送るイベントハンドラを document オブジェクトに設定する。また、サーバから HTML, CSS が送られたときに表示領域の HTML, CSS を書き換える。サーバの実装にはオープンソースで入手できるウェブレンダリングエンジンの中で、Safari, Google Chrome などにも使用されていて実績のある WebKit [3] を用いる。

## 5. おわりに

本研究では Web アプリケーションの学習を助けるための共有及び可視化手法と実装例を示した。基本的な Ajax Web アプリケーションであれば我々のシステムで支援可能である。

## 参考文献

- [1] D.Lowet , D.Goergen : Co-Browsing Dynamic Web Pages , WWW '09 Proceedings of the 18th International Conference on World Wide Web, pp.941-950.
- [2] 中村大介, 疋田輝雄 : 動的ウェブページのブラウザ共有, DICOMO 2010, pp. 1594-1604.
- [3]webkit.org : The WebKit Open Source Project. <http://www.webkit.org/>