

Webjig : ユーザ行動とユーザ画面の関連付けによる動的 Web サイト利用者の行動可視化システムの開発

木 浦 幹 雄^{†1} 大 平 雅 雄^{†1}
上 野 秀 剛^{†1} 松 本 健 一^{†1}

Webjig はサーバサイド/クライアントサイドで動的に処理される Web サイトに対応したユーザの行動を把握するためのシステムである。Web サイトのユーザビリティを向上させる為には、ユーザがどのように Web サイトを利用しているかを把握することが重要である。しかし、従来ツールでは Web サイト利用時のユーザの行動 (マウスカーソルの移動やマウスクリック等) を取得する際に、Web ブラウザに表示されている内容に着目しないため、サーバサイド/クライアントサイドで動的に処理される Web サイトを利用するユーザの行動を把握することが難しい。Webjig は、Web サイトの DOM を解析し、ユーザの Web ブラウザに表示されている内容と、ユーザ行動を関連付けて記録する。Webjig を利用することにより、サーバサイド/クライアントサイドで動的に処理される動的な Web サイトを利用するユーザの行動を正確に把握することが可能になり、効率的に Web サイトのユーザビリティを改善できる可能性がある。

Webjig: a visualization tool for analyzing user behaviors in dynamic web sites

MIKIO KIURA,^{†1} MASAO OHIRA,^{†1} HIDETAKE UWANO^{†1}
and KEN-ICHI MATSUMOTO ^{†1}

Visualizing user behaviors by recording a cursor motion and mouse click are important to improve web usability. It is, however, difficult for conventional visualization tools to track runtime user behaviors in a dynamically created web site because they do not analyze structures and designs of the web site. In this paper, we propose Webjig, a visualization tool for analyzing user behaviors in a dynamic web site. Webjig analyzes DOM (Document Object Model) of web sites, and maps user behaviors with contents displayed in a web browser. Webjig enables usability engineers to know exact user behaviors and helps them improve web usability efficiently.

1. はじめに

Web サイトのユーザビリティを向上させるためには、ユーザビリティ評価を行う必要がある⁹⁾。代表的なユーザビリティ評価の手法の 1 つとしてユーザビリティテストが挙げられる⁵⁾。古典的なユーザビリティテストでは、開発中の Web サイトを用いて、対象ユーザにユーザビリティ評価者が注目しているタスク (課題) を専用ルームで行ってもらう。この様子を観察したり、発話や操作内容を収集することによって、ユーザの思考過程や行動から設計の課題を明確にしたり、アイデアを発掘する。ユーザビリティテストを行うことにより、ユーザトラブル

を引き起こす重大な問題点や、開発者には思いもよらない問題点を発見しやすい¹⁰⁾。

しかし、古典的なユーザビリティテストを実施するためには多くのコストが必要となるため、容易に実施できるものではない³⁾。また、被験者が普段利用している環境を利用して評価することが出来ず、被験者にとって日常的でない状況で評価せざるを得ないことから、発見できない問題点が存在する可能性があると考えられる¹⁾。

そこで、ネットワークを介することによって、実際のユーザがどのように Web サイトを利用しているかを低コストで把握し、ユーザビリティの改善に役立てるためのシステムが提案されている^{1),2),6),7)}。しかし、これらの多くは静的な Web サイトを対象としているため、動的な Web サイトを利用するユーザの行動を正確に把握できないという問題点がある。なお、動的な Web サイトとは、下記のように

^{†1} 奈良先端科学技術大学院大学 情報科学研究科
Graduate School of Information Science, Nara Institute of
Science and Technology



(a) ドロップダウンメニューが閉じた状態



(b) ドロップダウンメニューが開いた状態

図 1 JavaScript によるドロップダウンメニューを採用した Web サイトの例

な Web サイトである。

- Web サーバ側で CGI やサーバサイドスクリプトによって動的に生成される Web サイト
- Web ブラウザ上で JavaScript によって実現される動的な Web サイト

例えば mixi や facebook 等に代表される SNS サイトは Web サーバ側で動的にページ内容を生成しており、同じ URL であったとしてもアクセスするタイミングやユーザ ID によって表示される画面が異なる。このため、同じ座標をクリックしたとしても、Web サイトの挙動が異なる場合がある。図を用いて具体例を示す。図 1 は JavaScript で実現したドロップダウンメニューを採用した Web サイトの例である。このサイトでは、ドロップダウンメニューが閉じている場合(図 1(a))と、開いている場合(図 1(b))では、同じ座標をクリックした際の挙動が異なる。しかし、マウスクリック位置の取得だけでは、ドロップダウンメニューが開いているか、閉じているか知ることが出来ないため、ユーザの行動を正確に把握することが出来ない。

他にも、JavaScript を用いることで、ページ遷移を発生させずにタブによる表示情報の切り替えや、ドラッグ&ドロップによるインターフェース等が実現可能である。このような動的な Web サイトにおけるユーザの行動をマウスカーソルの軌跡やクリック座標のみで把握することは非常に困難である。

本稿では、Web サイト利用時のユーザ行動とユーザ画面を関連付けて記録することによって、動的な Web サイトに対応したユーザ行動把握システムである Webjig について述べる。Webjig を利用することで、ユーザがどのように Web サイトを利用しているかをより詳細に把握することができるため、Web サイトのユーザビリティ向上につながると期待される。

2. 関連研究

従来、Web サーバのログを使用してユーザの行動を把握することで、Web サイトに潜むユーザビリティ上の問題点を発見し、ユーザビリティの改善に役立てる手法がとられてきた⁷⁾ Web サーバのログから、ユーザの IP アドレス、ユーザが Web サーバにアクセスした時間、ユーザが Web サーバにリクエストした内容、Web サーバがユーザに対して返した結果等を知ることが出来る。

Web サーバのログは自動的に保存され、安価に利用できるというメリットがある。しかし、下記のような問題点がある⁶⁾

- マウスカーソルの移動やクリック等のユーザ行動を把握することができない。
- Proxy サーバと動的な IP アドレス割り当てのため、データの信頼性が低い。

これらの問題点を解決し、Web サイトが実際のユーザからどのように利用されているかを把握するために、Proxy サーバを用いて Web ページに JavaScript コードを埋め込み、ユーザのマウスカーソルの移動やクリック等を収集するシステム (MouseTrack¹⁾, UsaProxy²⁾ 等) が提案されている。Web サーバのログに比べて、これらのシステムを利用することによって、ユーザが Web サイトをどのように利用しているかを詳細に把握することが可能である。

従来研究によって、Web 閲覧時のユーザの視線とマウスカーソルの位置に関する様々な知見が得られている。Chen らはマウスカーソルの位置と視線との間には強い相関があり、マウスカーソルの位置を把握することで、ユーザの興味がある箇所を予測し、ユーザの意図を推論できる可能性がある⁴⁾ さらに、Muller らの実験の結果、35% のユーザは Web ページの文章を読むとき、マウスカー

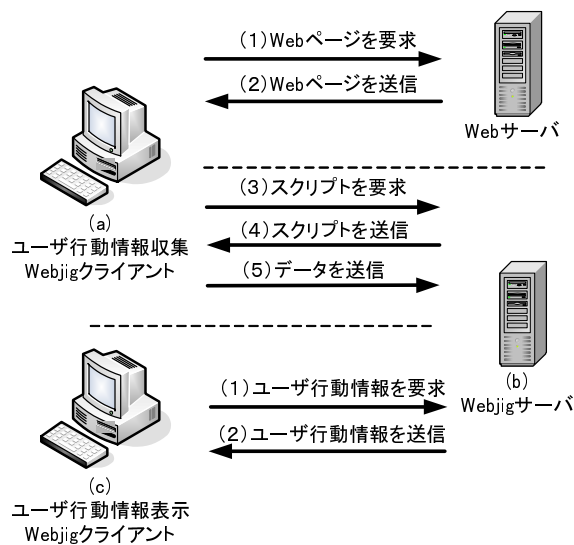


図 2 Webjig のシステム概要

ソルで文章をなぞることがわかっている⁸⁾。これらの事実
は、Web サイト使用時のユーザの行動を収集することが出
来れば、Web サイトのユーザビリティ上の問題点を発見で
きる可能性を示している。

しかし、MouseTrack や UsaProxy 等で収集するデータ
は、マウスカーソルの軌跡やクリック座標等に限られる。
このため、動的な Web サイト利用時のユーザ行動を把握
することが出来ないという問題がある。

3. Webjig の概要

我々は、従来ツールの「動的な Web サイトを利用する
ユーザの行動を把握できない」という問題点を解決するた
めのシステムとして Webjig を開発した。Webjig はマウス
カーソルの動きやマウスクリック等のユーザ行動情報と同
時に、ユーザ画面（ユーザが利用している Web ブラウザ
に表示されている内容）を取得することで、「動的な Web
サイトを利用するユーザの行動を把握できない」という問
題を解決する。

図 2 に、システムの概要を示す。Webjig はクライアン
ト-サーバシステムであり、クライアントは Web ブラウザ
上で動作する「ユーザ行動情報収集 Webjig クライアント」
(図 2(a)) は、Web サイトにアクセスしたユーザのユーザ行
動情報を Webjig サーバに送信する。「Webjig サーバ」(図
2(b)) は、ユーザ行動収集 Webjig クライアントから送信
された情報を解析し、処理しやすい形に加工した上でデー
タベースに格納する。「ユーザ行動情報表示 Webjig クライ
アント」(図 2(c)) は、Web サイトにアクセスしたユーザ
の行動を再生し、分析するためのソフトウェアである。以
下で、それぞれについて述べる。

3.1 ユーザ行動情報収集 Webjig クライアント

ユーザ行動情報収集 Webjig クライアントは JavaScript
で記述されている。Web サイトに埋め込むことによって、
Web ブラウザに読み込まれ、Web ブラウザ上で動作する。
このため、HTML テンプレートに外部 JavaScript を呼び
出すための Script タグの記述を追加するだけで、Web サ
イト利用時のユーザの行動を記録することが可能となる。

コードが挿入された Web サイトにユーザがアクセスし
たとき、システムは以下に示される順序で処理を行う(図
2 参照)

- (1) Web ブラウザが Web ページに対して HTTP リク
エストを送信する。
- (2) Web サーバは Web ブラウザに対して JavaScript
コードが挿入された Web ページを送信する。
- (3) Web ブラウザが Web サーバに対してクライアント
本体である JavaScript コードを要求する。
- (4) Web ブラウザは Web ブラウザに対して要求された
JavaScript コードを送信する。
- (5) Web ブラウザはユーザ行動情報の収集を開始し、収
集した情報を随時サーバに送信する。

なお、ユーザ行動情報には以下の情報が含まれる。

- ユーザ ID
- マウスのカーソルの位置
- マウスボタンの押下状態
- スクリーンサイズ
- スクロールバーの位置
- 表示している Web ページの DOM(Document Object
Model) 情報

ここでユーザ ID とは、Webjig が Web サイトにアクセ
スしたユーザを識別するための ID であり、Webjig サーバ
によって発行される。

3.2 ユーザ行動情報表示 Webjig クライアント

ユーザ行動情報表示 Webjig クライアントでは、Webjig
サーバに保存された情報から、Web サイト利用時のユーザ
の行動を再生することが出来る。

ユーザ行動情報表示 Webjig クライアントは、再生制御
ウィンドウ(図 3(a))と、ユーザ画面描画ウィンドウ(図
3(b))によって構成される。再生制御ウィンドウによって、
ユーザ行動の再生を制御することが可能である。また、ユー
ザ行動の再生中は常にユーザが Web サイトに滞在した総時
間と、Web サイトを訪問してからの経過時間を表示する。

また、通常の Web サイトは、複数の Web ページから構
成されていることが一般的であり、同一のユーザが複数の
Web ページに連続してアクセスした場合は、複数の Web
ページ内での行動を、連続的に再生することが出来る。ま
た、ページ内におけるマウスカーソルの軌跡を表示するこ

(b) ユーザ行動描画ウィンドウ



図 3 ユーザ行動情報表示 Webjig クライアントの動作イメージ

とが可能であるため、ユーザが Web ページ内のどこを見ているか、または見ていないかの判断を支援する。さらに、行動を時系列で表示する機能も備える。

3.3 Webjig サーバ

Webjig サーバでは、ユーザ行動情報収集及びユーザ行動情報表示クライアントの管理・制御と、データの管理を行う。ユーザがユーザ行動情報収集 Webjig クライアントが埋め込まれた Web サイトに初めてアクセスすると、Webjig サーバは、ユーザに対してユニークな ID を新規に発行する。この ID を用いることによって、複数のページ間における単一ユーザの行動を把握することが出来る。

また、Webjig サーバは各クライアントに対して命令する機能を持っており、たとえば、ユーザが特定の機能を使用している時だけデータを記録したり、特定のユーザの行動情報だけを収集する、などの制御が可能である。

さらに、各クライアントから送信されたデータを解析し、データベースに格納する機能を持つ。

4. ケーススタディ

本章では、JavaScript を使用している Web サイトを例に、Webjig を用いてユーザの行動を、どのように分析できるのかを示す。

対象とした Web サイトは奈良先端科学技術大学院大学の Web サイトのトップページ*1であり、ページ上部に配置されたメニューが JavaScript を用いて構成されている。

はじめに、ユーザの行動を把握するために、Web サイトを構成する HTML ファイルの末尾に、下記の 1 行を追加する。

```
<script type="text/javascript" src="ユーザ行動情報収集 Webjig クライアントを呼び出すためのアドレス"></script>
```

これは、HTML や JavaScript の知識を必要としないため、容易に作業可能である。

この JavaScript をユーザの Web ブラウザに読み込ませることで、ユーザの Web ブラウザで Webjig が実行され、ユーザ行動およびユーザ画面を記録することが可能である。

ユーザの行動を記録した後でユーザの行動を把握し、Web サイトの問題点を検討する作業を行う。Webjig では、ユーザの行動（マウスやキーボードの操作）とユーザが見ていた画面を同期して再生することが出来るため、そのユーザが、Web サイトのどこを見ていたのかに着目し、行動を分析することが可能である。また、ユーザ行動を再生中に一時停止や巻き戻しを行う事も可能であり、分析作業を効率的に進めることが出来る。

ユーザ行動情報表示 Webjig クライアントを用いてユーザ行動を再生した場合のイメージを図 3 に示す。再生制御ウィンドウ (図 3(a)) の再生ボタンを押下することによって、ユーザ行動描画ウィンドウ (図 3(b)) でユーザの行動を再生することが出来る。図 3(I) はユーザが Web サイトを訪問した直後の画面である。図 3(II) はユーザがドロップダウンメニューの上にマウスカーソルを置いた時点の画面である。そして、図 3(III) はユーザがドロップダウンメニューをたどり、項目を選択している画面である。このように、ユーザの行動とともに変化する画面の内容を同時に再生することが出来る。

従来のシステムの導入環境においてユーザが同じ行動を行った場合、ドロップダウンメニューを操作しても Web ページの遷移が発生せず、クライアント側で処理が行われ

*1 <http://www.naist.jp/index.j.html>

表 1 従来ツールとの比較

	Webjig	MouseTrack	UsaProxy
記録可能なユーザ行動	マウス操作等	マウス操作等	マウス操作等
動的なページへの対応	可	不可	不可
ユーザへの負担	なし	専用 Proxy を利用	Web ブラウザの設定変更
評価者への負担	小 (HTML に Script タグの埋め込み)	中 (ユーザに対して専用 Proxy の利用方法説明)	中 (ユーザに対して Web ブラウザの設定変更方法の説明)
ネットワーク負荷	大きい	中程度	中程度

るため、ドロップダウンメニューの変化を記録できない。したがって、図 3(I) の画面上でマウスカーソルだけが動くだけであり、ドロップダウンメニューを用いてどのような操作が行われたかを把握することが出来ない。Webjig では、ユーザの Web ブラウザに表示されていた内容と、ユーザの行動が同期して表示されることでユーザ行動を容易に把握することが可能である。

さらに、問題点を発見して Web サイトを改善すると、改善によってユーザの行動がどう変化するかを評価する作業が必要になる。Webjig では、ユーザ行動とユーザ画面を関連付けて記録するため、Web サイトの改善後と改善前で、ユーザ画面を確認しながらユーザ行動を比較することが可能である。

このように、Webjig を利用することで、動的な Web サイトのユーザビリティ評価を効率的に行うことが出来、Web サイトの問題発見及び、改善プロセスが容易になると考えられる。

5. 考 察

5.1 従来ツールとの比較

MouseTrack および UsaProxy は Webjig と同様に Web ページ内のユーザ行動に着目したシステムである。これらの 3 種類のツールの比較を表 1 に示す。

いずれのツールも、Web サイト利用時の、ユーザのマウスカーソルの移動や、マウスクリック等の行動を記録することが可能である。しかしながら、MouseTrack、UsaProxy では、ユーザが行動したときに、ユーザのブラウザにどういった画面が表示されていたかを記録することができない。Webjig は、JavaScript を用いて開発された動的な Web サイト利用時のユーザの行動を把握することが出来る。

また、MouseTrack および UsaProxy では、Web サイトを閲覧するユーザの行動を記録するためにユーザが自分自身で設定等を行わなければならない。MouseTrack では専用の Proxy を用いて Web サイトにアクセスする必要があり、UsaProxy では、ユーザの Web ブラウザの設定を変更する必要がある。Webjig では、ユーザはユーザ自身の環境に一切の変更を加える必要が無く、普段どおり Web サ

イトを利用することが出来る。

Webjig を用いてユーザ行動情報を収集するためには、Web サイトの HTML テンプレートにユーザ行動情報収集 Webjig クライアント用の JavaScript を読み込むための記述を追加する必要がある。しかし、この作業は HTML や JavaScript の専門知識を要求しないため、容易である。MouseTrack および UsaProxy では、Web サイトに対して作業を行う必要は無いが、ユーザに対して、専用 Proxy の使用方法や、ブラウザの設定変更などの方法を説明する必要がある。

いずれのツールもネットワークを介して動作するアプリケーションであり、ネットワークに対する負荷が発生する。MouseTrack および UsaProxy は、サーバに送信するデータは、マウスカーソルの動きやマウスクリック、キーの押下等に限られるためサイズが小さい。しかし、Webjig は、ユーザ画面を同時に記録し、サーバに送信するため、MouseTrack および UsaProxy と比較してデータサイズが大きく、ネットワークへの負荷が高い。そのため、ネットワーク負荷を低くすることが今後の課題である。

5.2 Webjig の応用

本節では Webjig の可能性について検討する。Webjig は遠隔 Web ユーザビリティテストを実施するために、Web サイト利用時のユーザの行動を把握するためのシステムとして開発を行った。「ユーザの行動と Web ブラウザに表示されている内容を関連付けて記録する」という特徴を利用することによって、次のように応用利用できる可能性がある。

5.2.1 ユーザのサポート

商用 Web サイトでは、電話等によるユーザサポートを提供している場合がある。しかし、商用 Web サイトには動的な Web サイトが多く、音声だけではユーザがどのような画面を見ているかを把握することが難しい。Webjig を利用することによって、ユーザ行動とユーザ画面を把握しながらサポートを行うことが出来るため、効率的なユーザサポートを提供することが可能になると考えられる。

5.2.2 再現性の無い不具合の追跡

開発中の Web サイトの HTML テンプレートに Webjig

を組み込むことによって、ユーザのすべての行動と、画面を記録することが出来る。このため、ユーザからの不具合報告をもとに、不具合が出現した前後に、ユーザがどのような操作を行ったかを調べて、把握することが出来る。これにより、Web サイトの不具合追跡が容易になると考えられる。

6. おわりに

本稿では、Web サイトの遠隔ユーザビリティテストを行うために、Web サイト利用時のユーザの行動を把握するシステム、Webjig を開発した。従来のシステムでは、マウスカーソルの軌跡やマウスクリック位置などのユーザの行動のみを記録していたため、動的な Web サイトのユーザ行動を把握することが出来なかった。Webjig では、Web サイト利用時のユーザ行動を、ユーザ画面と同時に記録することで、ユーザの行動を再現することが可能である。Webjig を活用することで、動的な Web サイトにおけるユーザの行動把握が容易になり、ユーザビリティの高い Web アプリケーションを開発することが容易になると考えられる。今後は、有用性を確認するための評価実験を実施することを検討している。

謝辞 本研究の一部は、独立行政法人 情報処理推進機構の 2008 年度上期 未踏 IT 人材発掘・育成事業「Web サイト閲覧中のユーザ行動を可視化する」による支援を受けて行われた。

参考文献

- 1) Arroyo, E., Selker, T. and Wei, W.: Usability tool for analysis of web designs using mouse tracks, *CHI '06 extended abstracts on Human factors in computing systems*, pp.484-489 (2006).
- 2) Atterer, R. and Schmidt, A.: Tracking the interaction of users with AJAX applications for usability testing, *the SIGCHI conference on Human factors in computing systems(CHI '07)*, pp.1347-1350 (2007).
- 3) Barnum, C. M.: *Usability Testing and Research*, Longman, London (2001).
- 4) Chen, M.C., Anderson, J.R. and Sohn, M.H.: What can a mouse cursor tell us more?: correlation of eye/mouse movements on web browsing, *CHI '01 extended abstracts on Human factors in computing systems*, pp.281-282 (2001).
- 5) Dumas, J.S. and Redish, J.C.: *A Practical Guide to Usability Testing*, Norwood, New Jersey, Ablex Publishing (1993).
- 6) Etgan, M. and Cantoe, J.: What does getting WET (Web Event-logging Tool) mean for

web usability?, *5th Conference on Human Factors and the Web(HFWEB99)*, available from <http://zing.ncsl.nist.gov/hfweb/proceedings/etgen-cantor/index.html> accessed 2008-10-10 (1999).

- 7) Hong, J.I. and Landay, J.A.: WebQuilt: a framework for capturing and visualizing the web experience, *the 10th international conference on World Wide Web(WWW '01)*, pp.717-724 (2001).
- 8) Mueller, F. and Lockerd, A.: Cheese: tracking mouse movement activity on websites, a tool for user modeling, *CHI '01 extended abstracts on Human factors in computing systems*, pp.279-280 (2001).
- 9) Nielsen, J. and Landauer, T.K.: A mathematical model of the finding of usability problems, *the INTERACT '93 and CHI '93 conference on Human factors in computing systems*, pp.206-213 (1993).
- 10) 岡田英彦：ユーザビリティとその評価手法，システム制御情報学会誌，Vol.45, No.5, pp.269-276 (2001).