

ユーザーの行動分析とアンケートの結果に基づく 自動的 Web コンテンツ配置システム

An Automatic Web Content Placement System Based on Users' Actions and Questionnaires

篠原 裕弥

Yuya Shinohara

法政大学情報科学部デジタルメディア学科

E-mail: yuya.shinohara.7b@stu.hosei.ac.jp

Abstract

In recent years, dynamic Web sites are increasing rapidly. Web developers are able to obtain users' actions in real time from dynamic Web sites. This makes it important especially to improve the usability of commercial Web sites. Although recent research partially enabled Web developers to understand users' behaviors in a dynamic Web site, page developers need to manually modify the site to improve its usability. This paper presents an automatic Web content placement system that uses users' actions in sites and questionnaires to improve Web sites. By using questionnaires, Web developers are able to understand users' needs. There are three advantages of this system. First, the system can improve the usability of Web sites without using developers' hands. Second, developers can understand which contents are actually important for users. Finally, it can keep the contents that match users' needs by hiding the other contents based on the results of questionnaires. The system consists of two main programs: one is a client program created in JavaScript; the other is a server program created with Node.js. To evaluate the effectiveness of the system, an experiment using a virtual Web site was conducted. Although the result shows the effectiveness of the system, it also indicates several problems.

1. はじめに

Ajax などの動的インターフェース作成手法の普及により急激に動的な Web サイトの数が増加した。動的インターフェースを持つ Web サイトではサーバー側と非同期で通信を行い、ユーザーの行動履歴をリアルタイムで取得することが可能である。取得した履歴をもとに、開発者が手動でユーザビリティを改善することが可能である。

動的サイトではフロントエンドに JavaScript 言語を用いて動的メニュー等を作成することが多い。これによってページ遷移をせずにコンテンツが移動、拡張するページを作成可能である。また、Web サイト上である一定のと

ころまでスクロールした段階で、以降のページを読み込むなどして、ページの処理速度を高速化し、ページの「重さ」を軽減することも可能である。このような利点から、現在では法人サイトや、ショッピングサイト、学校のサイトなど様々な Web サイトで動的インターフェースが用いられている。

動的インターフェースを持つ Web サイトにおいて、ユーザーが Web サイトのどこに注目しているかを確認することは困難とされてきていたが、近年の研究の成果により部分的に可能になってきている。しかし、行動分析をした結果に基づいて手動でページを変更しなければユーザビリティの改善に至ることは難しい。

本研究では非同期通信で取得したユーザーの行動履歴に基づいて決定した順位より Web サイト上のコンテンツを自動的に再配置することに加え、ユーザーに対する簡易ボタンアンケートを設置し回答させることでコンテンツの必要性をユーザーに評価させ、不要と判断されたコンテンツを非表示にすることで、ユーザーが目的のコンテンツに素早くアクセスできるようにする。自動的にページを変更しつづけることによって、ユーザーが使えるほどユーザビリティが改善するため、ページ開発者の負担が大幅に減ることが予想される。本研究の有用性を検証するため、仮想的に作成した Web サイトを用いての実験を行った。実験の結果、ユーザーがコンテンツにアクセスする時間を短縮することができた。しかし、ユーザーの能力により結果にバラつきが出ることなど問題点も残った。

2. 関連研究

Web サイトのユーザビリティに関する研究は複数存在する。1 つとして視線によるユーザーの行動分析に対する拡張表現の研究 [1]がある。この研究はユーザーによってページの表示が異なるため行動把握がしにくいなどの問題点がある。また、Marenkov らは自動的にユーザビリティを評価する Guideliner というシステムを開発、実験した [2]。あらかじめ作成したガイドラインに基づいて Web サイトの評価を自動的に行うことで開発者のユーザビリティ評価を容易にした。しかし、利用デバイスが限られるなどの問題点も残った。木浦らはユーザーの Web

ページ閲覧時の DOM (Document Object Model), マウスカーソルの軌跡等を取得する Webjig というシステムを作成, 利用することにより, 動的ページにおけるユーザーの行動分析をある程度可能にした [3]. この研究ではユーザーが最初に着目したカテゴリを追跡できるようにすることでユーザーの行動を分析するシステムを構築する. このシステムを用いたサイト利用実験により, ユーザビリティが向上していることが確認された. しかし, 有用性が示されたのは動的メニューのみであり, その他の動的コンテンツに対する有用性の検証はされていない.

近年, 商用でユーザーの行動分析に利用されているツールでヒートマップが存在する [4]. Web サイトにおけるヒートマップとは, マウスの動きなどのユーザーの行動の色によって可視化したものである. ヒートマップを用いることでユーザーが Web サイト利用時にサイトのどこに着目していたかを可視化することが可能であり, Web サイトのユーザビリティ改善に有用である.

しかし, これらの研究, 商用のツールにおいていずれも問題点となるのが「開発者が手動でサイトを変更しなければユーザビリティが改善しない」ことである. 本研究ではユーザーと開発者双方からユーザビリティに干渉でき, かつ, 自動的にユーザビリティが改善するシステムの開発を目的とする.

3. 提案手法

本研究ではユーザーが使えば使うほどページのユーザビリティを改善させるシステムの開発を目的として, 自動的コンテンツ再配置の手法を提案する. Web サイト上のユーザーの行動分析と各セクションの最下部に設置した簡易アンケートボタンを用いて手法を実現する. 行動分析の結果からユーザーがどれくらいコンテンツにアクセスしていたかを表す尺度を着目度, アンケートと行動記録の結果よりコンテンツがユーザーにとってどれくらい必要なものを表す尺度を重要度と定義し, 2 つの値とデフォルト値の計算により順位を決定する. 決定された順位により, ページ開発者の想定したページの主要な部分に順位の高いコンテンツを移動させる.

本手法には以下の 3 つの有用性がある. 1 つ目は自動的にページのコンテンツ移動を行うことによって, ページ開発者が手動でサイト改修を行う手間が少なくなることである. 2 つ目はページ下部に設置したコンテンツ等, ページ開発者があまり重要視していないコンテンツの中でアクセス数が多いものをページ上部に自動的に移動させることにより, Web サイトのユーザーがそのコンテンツにいち早くアクセスすることが可能になることである. 3 つ目は各セクションに設置された簡易アンケートに回答させ, 一定数否定的な回答がなされた場合, 該当のセクションを非表示にすることにより, ユーザーにとってより必要度の高い情報のみを Web サイトに残すことができることである.

3.1. ユーザーの行動記録

各コンテンツに対するクリックの回数, コンテンツ上のマウスの滞在時間, ユーザーのカーソルのスクロール

の座標などのユーザーの行動を記録し, データベースに保存する. 各コンテンツにあらかじめ着目度のデフォルト値を設定しておき, 一定時間以上マウスがコンテンツの上に滞在した場合などユーザーがコンテンツに対して着目していたと判断できる場合に値を上昇させる.

各コンテンツ i に対するクリックの履歴を x_i , マウスカーソルがコンテンツ上に一定秒数以上滞在した回数を y_i とすると以下の式で着目度を計算する.

$$\alpha x_i + \beta y_i$$

上記の式では重みとして α, β を用いた.

3.2. 簡易アンケートの実施

行動記録に加え, 各セクションに HTML の button タグで作成した簡易アンケートを設置し, ユーザーに回答させ, コンテンツの必要性を判断する. なお, アンケートについては強制的に答えさせるとユーザーが煩わしいと感じてしまうことが予想されるため, 回答は任意である. 各セクションの下部に設置したボタンをクリックすることにより, サーバー側に情報が送信されるように設計を行った. 重要度はデフォルト値を各コンテンツに対して設定し, アンケートの結果とユーザーの行動記録によりユーザーに対して不必要なコンテンツであると判断できる場合, 重要度の値を下降させる.

各コンテンツ i に対する重要度のデフォルト値を z_i , 「不必要なセクションだと感じる」ボタンをおした回数を γ_i とすると重要度は以下の式で表すことができる.

$$z_i - \gamma_i$$

3.3. コンテンツの再配置と非表示化

HTML 読み込み時にデータベースより行動の記録とアンケート結果を取得, 集計し, コンテンツごとに順位を決定する. コンテンツのクリック数, マウスのコンテンツ上の滞在時間, アンケートなどの結果により各セクションについてどれだけアクセスがあったか, ユーザーの着目度はどのくらい高かったかを基準として順位を決め, 決定された順位の順番でコンテンツをセクションごとに変更, 再配置する. 加えて, 前述のアンケートでユーザーの否定的な回答が一定数に達した場合, 該当するコンテンツの順位を順位外に設定し, コンテンツを非表示にすることによって, ユーザーにとって必要な情報のみが残るように Web ページを変更する. また, 新たなコンテンツを追加しようとした際は, ページ開発者は利用するセクション名をサーバー側, クライアント側の JavaScript ファイルに登録し, 新しく挿入するコンテンツを後述するセクションタグで括ることで利用できるように設計をしている.

4. 実装

本研究では, jQuery, MongoDB を用いて提案した機能を有したシステムを構築した. システムは様々なページに応用可能にするため, HTML に JavaScript ファイルを適用し利用できるようにする. また, 一般的なサイトに対応できるように, フレームワークは用いず, Node.js の基本的機能のみでサーバーサイドを構築した. 本研究のシス

テムでは Node.js で構築したローカルサーバーを用いて、実装・実験を行っている。

4.1. 行動履歴の記録

本研究ではユーザーの行動履歴として主に 3 つを取得し、集計する。1 つは Web サイト上のコンテンツをクリックした回数である。このクリック回数を MongoDB の clickSection というコレクションに保存後、集計する。集計したデータをもとに click.json という JSON ファイルを作成し、着目度の計算に利用する。2 つ目は Web サイト上の特定のコンテンツに対して一定秒数以上マウスが上に滞在した場合、その滞在したコンテンツのセクション名を取得するものである。取得したセクション名を stopSection という MongoDB 上に作成したコレクションに保存し、集計する。なお、秒数はページ開発者が自由に設定できるよう、プログラム上では変数に格納している。集計した結果をもとに stop.json という JSON ファイルを作成し、計算に利用する。3 つ目はスクロール位置の取得である。これは上記の 2 つを取得する際に同時に取得し、ページ内でのマウスカーソルの位置を保存するものである。

4.2. アンケートの記録

アンケートは JavaScript で作成したボタンを用いて取得をする。「不必要なセクションだと感じる」という文字の入ったボタンを作成し(図 1)、ユーザーがボタンをクリックしたらサーバー側にクリックされたセクション名が post 送信されるように設計を行った。サーバー側は送られてきたセクション名を取得し JSON に変換後、MongoDB に作成した questionnaire というコレクションに結果を保存する。保存された結果をもとに questionnaire.json という JSON ファイルを作成し、作成された JSON ファイルをクライアント側で読み込みコンテンツを非表示にする。一定数ユーザーに「不必要」と回答されたコンテンツを非表示にするために利用する値は開発者側で設定できる。開発者側で設定した値をアンケート結果が上回った場合、コンテンツを非表示にするよう JavaScript で設定を行った。



図 1. アンケート用ボタン

4.3. システム導入の必須・推奨事項

本研究で作成したシステムを用いるために jQuery の導入を必須としている。システムの再配置で jQuery オブジェクトを用いてセクションを指定しているため、導入元の Web サイトの HTML に jQuery へのリンクが張られていることを必須条件とする。また、導入時にはサーバー

側、クライアント側それぞれの JavaScript ファイル内に変数として設定した sections という配列内に jQuery オブジェクトの名前を登録することでユーザーの行動履歴の取得、保存、利用を可能としている。Web サイトによって特殊なフレームワークやセクション名を設定していることが多くの Web サイトにおいて想定されるため、配置を変更したいセクションを HTML の div タグもしくは section タグで囲い、id に「section1」などのように「section」と通し番号を組み合わせたものを設定することを推奨する。

5. 実験

本研究の手法の有用性を検討するために被験者実験を行った。本実験ではあらかじめ作成した仮想的なコーヒー豆の紹介 Web サイトを用いた。本研究の有用性を検証できるようにするために、セクションの数の多い Web サイトを作成した(図 2)。本サイトは 2 冊の書籍 [5] [6] を参考にしている。作成に CSS のフレームワークである Bootstrap を用いることで、レスポンス性の高いサイトを実現した。



図 2. 実験用仮想 Web サイト

5.1. 被験者とタスク

被験者 8 名(男性 7 名, 女性 1 名, 平均年齢 24.7 歳)を対象にして実験を行った。8 名を 2 グループに分け、前半の 4 名で仮想の Web サイトを用いて行動履歴を取得する。後半の 4 名は、前半で取得した行動履歴に基づきコンテンツの再配置を行った後の Web サイトで実験を行う。実験のタスクとして、仮想 Web サイト上で事前に設定した世界各地のコーヒー豆の品種や精製方法、フレーバーなどを探し、解答してもらう。解答は事前に用意した紙に手書きで解答してもらう。Web サイトには 19 個のセクションを設置しているため、それぞれのセクションから 1 つずつコーヒー豆の詳細についての質問を作成し、ランダムに 13 個の問題を選び出し、その質問にすべて解答し終わるまでの時間を測る。時間の計測はストップウォッチで行った。13 個の問題に解答後、正答率を計算する。実験終了後、被験者に対し、システムがあることによってページが使いやすくなっているか、システムの改善点があるか等のアンケートを実施する。前半の 4 名の行動履歴より、Web ページの配置を変更し、再度前半の 4 名と同じ実験を後半の 4 名に行う。なお、後半の 4 名に関して、行動履歴は取得せず、アンケート機能のみを用いて実験を行う。

5.2. 被験者への指示

本研究の実験では13個の質問に答えるにあたって、必要がないと判断できるセクションに対して不必要であるというボタンを押してもらっている。必要がないセクションというのはセクション群の中で質問に登場しないものである。また、実験を行う前にあらかじめコーヒー豆の詳細ページのデザインと同じサンプルページを見てもらい、質問の意味を理解してもらっている。

5.3. 実験環境

実験ではデバイスや通信環境の違いにより差が出ないようにするため環境を統一した。PCはLet's note CF-SZ5のCore i7モデルで、通信はiPhoneによるSoftBankのデザリングで行った。ポインティングデバイスはマウスを使用し、WebブラウザはFirefoxを使用した。

5.4. 実験結果

被験者8名の実験にかかった秒数、正答率、問題すべてを解答し終えるまでにかかった秒数を解答数で割った秒数を表1に示す。被験者1~4がコンテンツの再配置前のWebサイトで実験を行ったもの、被験者5~8が再配置後に実験を行ったものである。

表1 実験結果

被験者	秒数	正答率	秒数/解答数
1	13分17秒	100% (13問)	61.29秒
2	15分02秒	100% (13問)	69.36秒
3	12分54秒	96% (13問)	59.52秒
4	12分23秒	100% (13問)	57.12秒
5	7分16秒	100% (13問)	33.46秒
6	5分24秒	100% (13問)	24.85秒
7	5分25秒	100% (13問)	25.00秒
8	6分35秒	100% (13問)	30.31秒

正答率の括弧の中身は解答できた問題数を示している。1問解答するのにかかった時間の平均は、システムによって再配置を行っていない前半4名で61.82秒、再配置後の後半4名で28.41秒であり、分散はそれぞれ28.17, 17.80であった。秒数/解答数の結果に対してt検定を行った結果、有意差が認められた($p < 0.05$)。実験後に行ったアンケートで以下の意見が得られた。

- ・不必要なセクションだと感じづらい
- ・実験方法を理解するのに時間がかかる
- ・サイト内検索が欲しい
- ・どこを開いたのか確認できるようにしてほしい
- ・実験用サイトに番号が欲しい
- ・おすすみが表示されて欲しい

「サイト内検索が欲しい」という意見については5名の被験者からの要望があった。

6. 議論

実験では本研究の目的通り、コンテンツにたどり着く時間を短縮することができた。また、すべての被験者が全質問に解答できたことから、簡易アンケートが正常に機能していたことも確認できた。しかし、実験を行うにあたってわかった問題点が複数存在する。1つ目として、

実験用のWebサイトが煩雑ではなく、時間をかけずに目的のコンテンツにたどり着けたことである。2つ目に、システムの運用時間が短いことである。より複雑なサイトで長期間システムを運用すれば実用的かどうかの判断がより正確に下せるようになると考えられる。最後に、ユーザーによっては、システム開発者が意図していない行動をとり、目的のコンテンツが消える可能性があることである。

簡易アンケートボタンはユーザーが普段見かけないものであるため、どのように使ったらよいかかわからないという意見も多かった。実験では事前に説明を行ってWebサイトを使ってもらっているため、ボタンを押してくれる被験者も多かったが、一般の説明を受けていないユーザーにとっては使いどころがわからないボタンになると考えられる。ただし、使い方が理解できればボタンを押す被験者も多かったため、他の機能を持った簡易アンケートボタンを設置してもよいと考えられる。

2秒以上滞在した場合にそのコンテンツを取得し、post送信後データベースに保存する機能については、実験の結果、保存数が想定よりも少ないことが確認できた。これは2秒という設定が長すぎたことを示していると考えられる。ユーザーがどのコンテンツに着目しているかより詳細に確認したい場合、サーバーへの負荷は大きくなるが、より短い時間に設定した方がよいと考えられる。

7. おわりに

本研究では、使えば使うほどユーザビリティを改善するシステムを目的として開発を行った。本研究で作成したシステムによって目的のコンテンツにたどり着くまでの時間を短縮できることを確認した。同時に様々な問題点が残ることも明らかになった。ユーザーの行動を制限する機能の実装等、改善の余地があると考えられる。

文 献

- [1] R. Menges, H. Tamimi, C. Kumar, T. Walber, C. Schaefer and S. Staab, "Enhanced Representation of Web Pages for Usability Analysis with Eye Tracking," *Proc. ETRA'18*, no. 18, pp. 1-9, 2018.
- [2] J. Marenkov, T. Robal and A. Kajia, "Guideliner – a Tool to Improve Web UI Development for Better Usability," *Proc. WIMS'18*, no. 17, pp. 1-9, 2018.
- [3] 木浦幹雄, 大平雅雄, 上野秀剛, 松本健一, "Webjig : ユーザー行動とユーザー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及び評価," vol. 51, no. 1, pp. 204-215, 2010.
- [4] 株式会社ユーザーローカル, "無料ヒートマップ解析ツール User Heat : どこが読まれているか見えるアクセス解析," 2019. [Online]. Available: <https://userheat.com/>.
- [5] 藤田政雄, 知れば知るほどおいしく飲めるコーヒー辞典, 日本文芸社, 2010.
- [6] 若松和紀, 極める愉しむ珈琲辞典, 西東社, 2017.