

仮想DOMを用いたWebページ双方向編集システム

若松 祐樹^{1,a)} 早川 智一^{2,b)} 疋田 輝雄^{2,c)}

概要: Web ページを閲覧者と開発者が双方向で編集できるシステムを提案する。提案システムの目的は、開発者支援のための Web ページデザイン情報を閲覧者から取得することにある。各閲覧者は、Web ページのデザインを自分好みに編集でき、開発者は各閲覧者が編集した Web ページのデザインを確認しつつ Web ページの開発が行える。提案システムでは、HTML 要素の位置の移動や削除、文章のスタイル編集などが Web ブラウザ上で行える。また、閲覧者と開発者が生成した異なる 2 つのデザイン情報を木構造のオブジェクトとし、それらの差分を抽出・併合する際に仮想 DOM の概念を用いることで、両者が互いを意識せずに編集することを可能にする。我々は、両者が双方向編集した際にデザインが崩れないかを複数の Web ブラウザと代表的な Web ページとの組合せにおいて評価し、それら組合せの 90% でデザインが崩れないことを確認した。

1. はじめに

Web ページを開発する上でデザイン（レイアウトやスタイル）は、Web ページのアクセシビリティやユーザビリティに関わる重要な要素である。アクセシビリティとは、Web ページがどの程度のユーザや環境（OS や Web ブラウザなど）に適用するかを示すもので、ユーザビリティとは、Web ページの使いやすさを示すものである。

アクセシビリティやユーザビリティを向上させるために Web ページ開発者（以下、開発者）が Web ページ開発（特にデザインの修正）を行う際には、Web ページ閲覧者（以下、閲覧者）からの Web ページデザインに対するフィードバック情報を考慮することが望ましい。ただし、Web ページ内での閲覧者の行動履歴をフィードバックとして取得し、それによりデザイン上の問題点を発見し修正するというのが一般的な Web ページ開発の方法である。

閲覧者からのフィードバック情報は、開発者にとって Web ページ開発のために有用であるが、閲覧者がフィードバックを行ったとしても、それに対する結果（主にデザイン）が閲覧者にとって必ずしも満足できるものとは限らない。なぜならば、すべての閲覧者の好みに合うデザインを考案するのは現実的ではないからである。

もし、フィードバックを行った閲覧者に対して満足度の高い Web ページのデザインを提供できれば、フィードバックする閲覧者は増加し、より多くのフィードバック情報を取得できると考える。また、Web ページのデザインに関するフィードバックを閲覧者が直接できれば、デザインを修正するためのより有用な情報を開発者が取得できると考える。

本研究の目的は、閲覧者が Web ページのデザインを自分好みに編集し、そのデザインを開発者に容易にフィードバックできるようにすることで、開発者が Web ページの開発を行う際に有用となる Web ページデザイン情報（以下、デザイン情報）を取得することにある。Web ページ開発のためのデザイン情報を閲覧者から取得することで、Web ページのデザインを改善するための判断材料が増えるので、開発者は、より閲覧者の意見を取り入れた質の高いデザインを Web ページに施すことができると我々は考える。

我々は、閲覧者からのフィードバックを促し、閲覧者からのデザイン情報を開発者が得るための方法として Web ページ双方向編集システム（以下、W-style）を提案し、その有用性を評価した。なお、Web ページのデザインを閲覧者ごとに編集できる機能が閲覧者にとって有用であることは、我々の先行研究 [12] で示している。

2 節では関連する技術や論文について言及する。3 節では提案手法について説明する。4 節と 5 節とでは W-style の設計と実装について詳説する。6 節では W-style の評価結果について報告する。7 節ではまとめと今後の展望について述べる。

¹ 明治大学大学院理工学研究科
School of Science and Technology, Meiji University

² 明治大学理工学部
School of Science and Technology, Meiji University

a) wakamatsu@cs.meiji.ac.jp

b) t_haya@cs.meiji.ac.jp

c) hikita@cs.meiji.ac.jp

2. 関連研究

2.1 関連技術

Wiki は、Web ブラウザ上から Web ページの編集を行うことができる CMS (Content Management System) の一つである。W-style とは、閲覧者が Web ブラウザ上で Web ページを編集できる点で類似しているが、Web ページの表示方法に違いがある。Wiki では 1 人が編集した内容をすべての閲覧者が共有するが、W-style では編集した内容は編集した閲覧者のみに適用される。

Stylebot[1] は、閲覧中の Web ページのスタイルを閲覧者好みに編集できる Google Chrome の拡張機能である。W-style とは、閲覧者が Web ページを自分好みに編集できるという点で類似しているが、フィードバックの有無に違いがある。Stylebot では編集したデザイン情報を開発者にフィードバックできないが、W-style では開発者にフィードバックできる。

Wishbox[3] は、閲覧中の Web ページのスクリーンショットを撮り、その画像にコメントや図を描画してメール送信ができるフィードバック用の問い合わせフォームである。W-style とは、閲覧者が能動的に Web ページに対するフィードバックを送信できる点で類似しているが、フィードバック送信後の Web ページのデザインに違いがある。Wishbox では、Web ページのデザインを閲覧者が編集できないため開発者が編集するまでデザインが変わらないが、W-style では、閲覧者ごとに Web ページのデザインを編集できるため開発者が編集しなくてもデザインが変更できる。

2.2 関連論文

木浦ら [10] は Webjig というシステムを提案している。Webjig と W-style とでは、Web ページに対する閲覧者の行動を開発者が情報として取得できるという点で類似しているが、取得する情報が異なる。Webjig では、Web ページ中での閲覧者の操作履歴と動的に変化する表示内容を取得するが、W-style では、閲覧者が編集した Web ページのデザインを取得する。また、Webjig では閲覧者は通常のブラウジングを行えるが、W-style では通常のブラウジングに加えてデザインの編集が行える点にも違いがある。

3. 提案手法

3.1 提供する機能

W-style では、Web ページにデザイン編集用の機能 (ボタンなど) を新たに追加することで、閲覧者が

- (1) Web ブラウザ上で Web ページのデザインを編集し、
- (2) 編集したデザイン情報を保存することを可能にする。(1)により、閲覧者が専用のソフトや

プラグインなどをインストールする手間を省くことができる。(2)では、編集したデザイン情報が閲覧者の Web ブラウザ上と開発者の Web サーバ上とに保存される。デザイン情報を Web ブラウザ上に保存することで、編集した Web ページのデザインを他の閲覧者と共有せずに、同一の Web ページに対して異なるデザインを閲覧者ごとに提供できる。また、開発者の Web サーバ上にもデザイン情報を保存 (フィードバック) することで、開発者は閲覧者がどのようにデザインを編集したのかを知ることができる。開発者は、閲覧者から得たデザイン情報を基に、閲覧者にとってより満足度の高いデザインに Web ページを編集できる。

3.2 WYSIWYG の採用

W-style による Web ページの編集は、WYSIWYG (What You See Is What You Get) の概念を用いて実現する。WYSIWYG を採用した理由は、閲覧者が W-style を使用する際の学習コストを低減できると考えたためである。WYSIWYG を用いた編集方法を取り入れることにより、HTML (HyperText Markup Language) や XHTML (Extensible HyperText Markup Language)、CSS (Cascading Style Sheets) などの専門的な知識を必要とせずに、マウスとキーボード操作のみで Web ページの編集が可能になるため、それらを学ぶための学習コストを削減することができる。

3.3 仮想 DOM の採用

W-style では、Web ブラウザに Web ページを表示するための DOM (Document Object Model) [7] を構築する際に仮想 DOM の概念を用いる。仮想 DOM とは、生の DOM とは異なる別の木構造のオブジェクトのことである。仮想 DOM を用いて 2 つの DOM の構造の差異を算出し、それに対する適切な処理を行うことで DOM の操作を最小にすることができる。

仮想 DOM を採用した理由は、W-style の機能を実現する際に、2 つの異なる構造をした DOM を 1 つに併合し、新たな構造の DOM を構築する必要があるためである。W-style では、閲覧者と開発者との双方が Web ページを編集できる仕組みのため、Web ページを Web ブラウザ上に描画する前に、それぞれの編集したデザイン情報を併合して新たな DOM を構築する必要がある。

3.4 W-style の実行例

図 2、図 3、図 4 は、Amazon のトップページ (図 1) に W-style を利用してデザインの編集をした一例である。図 2 では、画面上側と右側の HTML 要素を閲覧者が削除し、閲覧者のローカル環境の画像を背景に適用している。このデザインは、開発者が編集しておらず閲覧者が編集し

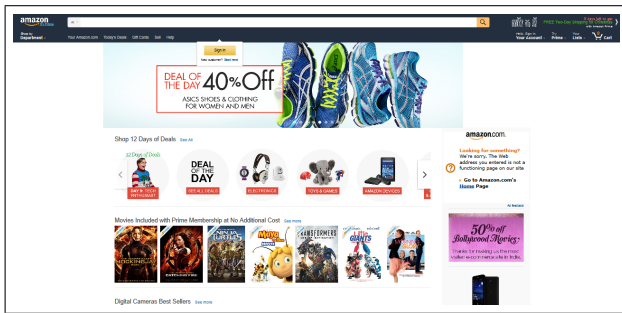


図 1 W-style を利用してデザインの編集を行う前の Amazon のトップページ

Fig. 1 Amazon top page before editing the design by using W-style.

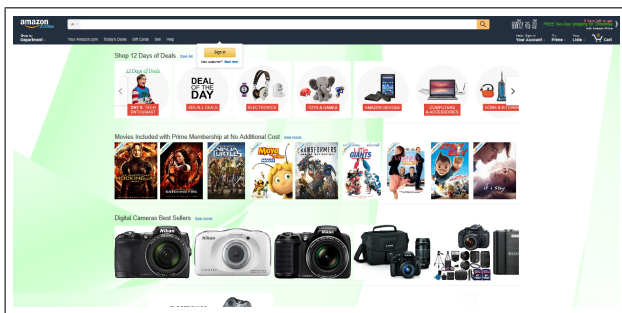


図 2 W-style を利用して図 1 のデザインを閲覧者が編集した例
Fig. 2 Example of viewer's editing Fig.1 design by using W-style.

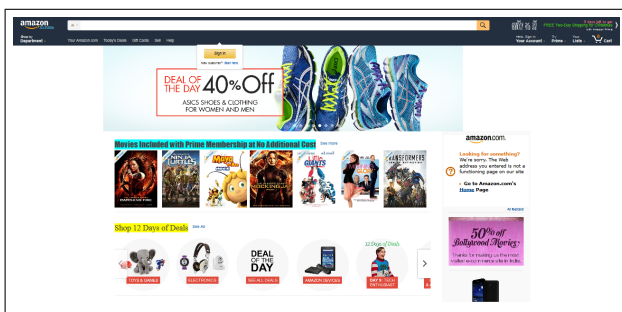


図 3 W-style を利用して図 1 のデザインを開発者が編集した例
Fig. 3 Example of developer's editing Fig.1 design by using W-style.

ている場合に、編集した閲覧者にのみ適用される。図 3 では、画面下半分のいくつかの HTML 要素の位置を変更し、文字の背景色・フォント・大きさを編集している。このデザインは、閲覧者が編集しておらず開発者が編集している場合に、すべての閲覧者に適用される。図 4 は、図 2 と図 3 とのデザインを併合させた画面である。このデザインは、閲覧者と開発者がとが編集している場合に、図 2 のデザインに編集した閲覧者にのみ適用される。

4. W-style の設計

4.1 前提

W-style は、複数の Web ブラウザに対応するクロスブラ

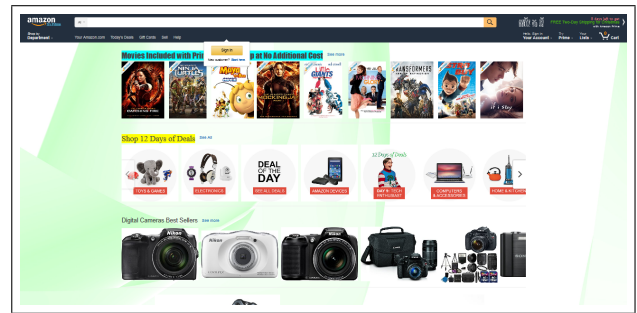


図 4 図 2 と図 3 のデザインを併合した Web ページの例
Fig. 4 Example of web page merged from Fig.2 and Fig.3.

ウザを実現する。なお、Web ページ内の文章を選択する際に、一度に選択できる範囲が Web ブラウザによって異なるが、文章の内容や装飾の編集に関してはすべてのブラウザで一樣に動作する。また、初めから Web ページのデザインが Web ブラウザ毎に異なるものである場合、デザインを併合する際に Web ブラウザによってはデザインが崩れる可能性がある。

W-style は、HTML5[8] 準拠の Web ブラウザでの利用を前提としている。これは、編集機能を実現する際に、HTML5 で新たに追加・標準化された API (Application Programming Interface) を使用しているためである。

W-style により Web ページの編集を行う際には、他の Web ページ開発システムとの併用をしないこととする。W-style では、2つの異なる DOM を比較する必要があるため、それぞれの DOM の整合性を保つ必要がある。他のシステムを併用すると、DOM の整合性を保つことが困難となるため、Web ページの編集は W-style のみで行うことを前提とする。

開発者が Web ページに W-style を導入する場合は、

- (1) Web ページが公開されている Web サーバに、W-style を実現するための専用のファイル群 (HTML, JavaScript ファイルなど) をインストールし、
- (2) インストールした JavaScript ファイルを参照する `script` タグを HTML 中の `head` 要素内に 1 行だけ記述する

必要がある。一方で、閲覧者が W-style を利用する際、特別な処理 (専用プラグインのインストールなど) は必要としない。

4.2 W-style の動作

W-style のシーケンス図を図 5 に示す。この図が表す動作の流れは以下の通りである。

- (1) 利用者が開発者であれば編集用の Web ページにアクセスする。
- (2) 利用者が閲覧者であれば閲覧したい Web ページに HTTP 要求し、開発者であれば編集用 Web ページの中の URL 入力フォームに編集したい Web ページの

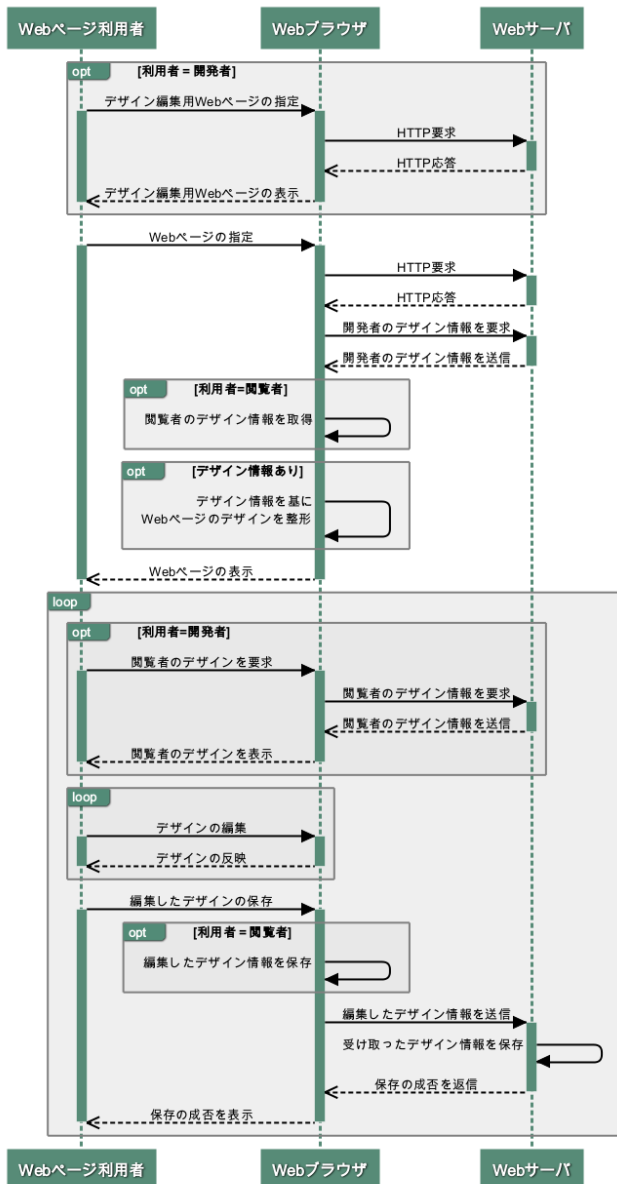


図 5 W-style のシーケンス図

Fig. 5 Sequence diagram of using W-style.

URL を入力し HTTP 要求をする。

- (3) Web ブラウザは、HTTP 応答を取得後、保存されているデザイン情報を Web サーバから（閲覧者の場合は Web ブラウザからも）取得する。
- (4) Web ブラウザは、デザイン情報が存在する場合、それを基に Web ページのデザインを整形する。
- (5) Web ブラウザ上に Web ページを表示する。
- (6) 利用者が開発者であれば閲覧者が編集したデザインを編集用 Web ページから確認することができる。
- (7) 利用者は、Web ブラウザ上で Web ページのデザインの編集を行う。
- (8) 編集したデザインの保存は、利用者が閲覧者であれば Web ブラウザと Web サーバに、開発者であれば Web サーバのみに行う。

4.3 提供する機能

W-style は、閲覧者が Web ページ中の

- HTML 要素の位置を移動,
- HTML 要素の大きさを変更,
- HTML 要素を削除,
- HTML 要素の背景の色や画像を変更,
- HTML 要素としてローカル環境の画像を挿入

する機能を提供する。また開発者には、閲覧者に提供する機能に加え、Web ページ中の

- テキスト（スタイルを含む）を編集,
 - テキストにハイパーリンクを設定,
 - HTML 要素として新しいテキストボックスを追加
- する機能を提供する。

5. W-style の実装

5.1 使用技術

W-style は、クライアント側を JavaScript で、サーバ側を PHP で実装した Web システムである。これらのプログラミング言語を採用した理由は、クライアント言語とサーバ言語のそれぞれで最も利用されている [4][5] からである。これにより、プログラミング言語の学習コストやシステムの開発・運用・保守のコストを低減できる。なお、クライアント言語は JavaScript で固定であるが、サーバ言語は PHP 以外の言語に変更が可能である。

5.2 実装上の特徴

W-style を実装する上で、特徴的な点を以下に示す：

Web ページのデザインを編集してもオリジナルの HTML ファイルは変更しない。Web ブラウザに HTTP 応答で返ってくる HTML データは閲覧者と開発者とのどちらの場合でも常に同じで、Web ページにアクセスする度に、保存されているデザイン情報に応じてデザインの整形を行う。

HTML 要素を識別するための独自属性を付与する。1 つ目に、Web ページにアクセスする度に body 要素を含む子孫要素すべてに data-id 属性を付与する。data-id 属性には、DOM 内で一意になるように HTML 要素ごとに異なる値を設定する。2 つ目に、閲覧者の場合に限り、Web ページのデザインを編集する時に操作した要素に data-edit 属性を付与する。data-edit 属性の値には、要素のスタイルを編集した際は“style”の文字を、要素の位置を変更した際には“move”の文字を設定する。両方を編集した場合は、“stylemove”の文字を設定する。

閲覧者や開発者が編集したデザインを保存するとき、body 要素の outerHTML をデザイン情報として保存する。これにより、data-id 属性を保持したままデザイン情報を保存することができる。

5.3 デザインの併合

W-style では、閲覧者が編集したデザイン情報と開発者が編集したデザイン情報とを併合して Web ブラウザ上に描画する必要があり、その際に仮想 DOM の概念を使用する。独自属性を付与した異なる 2 つの構造を持つ DOM (オリジナルの DOM と仮想 DOM) を併合する際の DOM の変化の様子を図 6 に示す。デザインの併合の流れは以下の通りである。

- (1) Web サーバから開発者のデザイン情報を、Web ブラウザから閲覧者のデザイン情報を取得する。
- (2) オリジナルの DOM のクローン (以下、仮想 DOM) を生成する。
- (3) オリジナルの DOM の body 要素の outerHTML と開発者のデザイン情報とを置換する。
- (4) 仮想 DOM の body 要素の outerHTML と閲覧者のデザイン情報とを置換する。
- (5) オリジナルの DOM の body 要素の全子孫要素を参照する配列 (以下、オリジナル配列) を取得する。
- (6) 仮想 DOM の body 要素の全子孫要素を参照する配列 (以下、仮想配列) を取得する。
- (7) 2 つの配列を走査し、オリジナルの DOM にあって仮想 DOM にない data-id 属性の値を持つ HTML 要素をオリジナルの DOM から削除する。
- (8) 2 つの配列を走査し、仮想 DOM にあってオリジナルの DOM にない data-id 属性の値を持つ HTML 要素をオリジナルの DOM に追加する。
- (9) 仮想配列を先頭から走査し、data-edit 属性が付与された HTML 要素 (以下、仮要素) を探す。
- (10) オリジナル配列を先頭から走査し、仮要素と同じ data-id を持つ HTML 要素 (以下、オリジナル要素) を探す。
- (11) 仮要素の data-edit 属性の値に “style” の文字があれば、オリジナル要素の style 属性に仮要素と同じ値の style 属性を設定する。
- (12) 仮要素の data-edit 属性の値に “move” の文字があれば、オリジナル要素の (兄弟要素内での) 位置を仮要素の (兄弟要素内での) 位置と同じにする。
- (13) (9) から (12) までの処理を data-edit 属性が付与された要素の数だけ行う。

DOM の書き換えを行う際に、閲覧者と開発者が同一の HTML 要素に異なる編集を行っていた場合、閲覧者のデザインを優先するように DOM を書き換える。これは、閲覧者が編集した後のデザインが開発者に上書きされると、閲覧者のユーザビリティが低下すると考えたからである。

6. 評価

我々は、W-style の有用性を評価するために、閲覧者と

表 1 世界シェアの上位 5 ブラウザ

Table 1 Top 5 web browsers currently used in the world.

Web ブラウザ	バージョン
Internet Explorer	11.0
Mozilla Firefox	42.0
Google Chrome	47.0
Safari	9.0
Opera	34.0

開発者とが Web ページを双方向編集した際にデザインが崩れないかを、複数の Web ブラウザと代表的な Web ページとの組合せで検証した。評価に使用した Web ブラウザを表 1 に、Web サイトを表 2 に示す。Web ブラウザは世界シェアの上位 5 ブラウザ [6] を、Web ページは Web トラフィック量の上位 10 サイト [2] の各トップページをそれぞれ評価対象とした。

我々は、W-style が Web ページのデザインの崩れを引き起こさないことを確認するために、W-style で編集した Web ページのデザインと、Web ブラウザに標準搭載されている Web 開発機能で編集した Web ページのデザインとを比較した。このとき実施した編集作業は、(1) HTML 要素の位置の移動、(2) HTML 要素の削除、(3) 背景画像の変更、(4) 文章の編集の 4 つとした。なお、W-style では (2) と (3) を閲覧者が、(1) と (4) を開発者が行い、それぞれが編集したデザインを併合したものを比較に使用した。

それぞれの Web ページのデザインを比較するために、デザイン編集後の Web ページのスクリーンショットを撮影し、それらの画像の類似度を SSIM[9] で測定した。SSIM とは画像評価を行う方法の 1 つで、他の方法に比べ、人間が感じる違いの指標をより正確に測定できるため採用した。

SSIM での画像評価の結果を表 3 に示す。数値が 1 であれば比較した 2 つの画像が同じであることを表し、数値が低くなるにつれて差分が大きいことを表す。数値が 0.98 以上であれば比較した画像がほとんど同じと言える [11] ことから、W-style は表 1 と表 2 の組合せの 50 通りの内の 45 通り (90%) でデザインの崩れを引き起こさないことが確認できた。

我々は、評価の結果から、W-style がデザインの崩れを引き起こすことなく、閲覧者と開発者とが Web ページを双方向編集できるという結論を得た。

7. おわりに

本論文では、Web ページのデザインに対する閲覧者からの Web ページデザインに対するフィードバックを容易に可能にし、開発者が Web ページの開発を行う際に有用となり得るデザイン情報を取得する方法として W-style を提案した。評価の結果から、W-style は Web ページのデザインを崩すことなく閲覧者と開発者とが双方向で編集できる

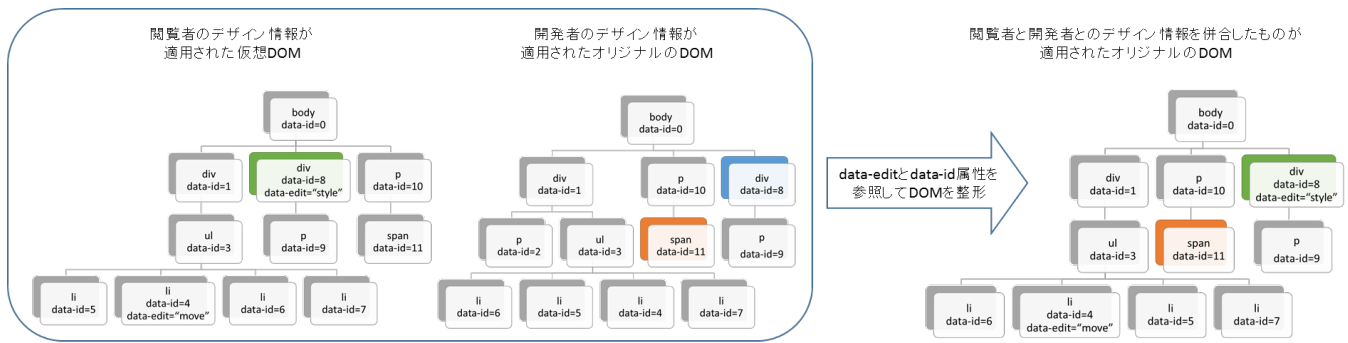


図 6 独自の属性を付与した異なる構造を持つ 2 つの DOM の併合例
Fig. 6 Merging example of two DOMs with different attributes.

表 2 Web トラフィック量の上位 10 サイト
Table 2 Top 10 high-traffic web sites.

Web サイト	URL
1 Google.com	http://google.com/
2 Facebook.com	http://facebook.com/
3 YouTube.com	http://youtube.com/
4 Baidu.com	http://baidu.com/
5 Amazon.com	http://amazon.com/
6 Yahoo.com	http://yahoo.com/
7 Wikipedia.org	http://wikipedia.org/
8 Qq.com	http://qq.com/
9 Twitter.com	http://twitter.com/
10 Google.co.in	http://google.co.in/

表 3 W-style と Web ブラウザの開発機能とでデザインを編集した各 Web ページのスクリーンショットの類似度 (SSIM 値)

Table 3 SSIM values of web page screenshots for all combinations of web sites and browsers.

Web サイト	A	B	C	D	E
Google.com	0.999	0.999	0.999	0.999	0.999
Facebook.com	1.000	1.000	1.000	1.000	1.000
YouTube.com	0.648	0.943	0.950	0.942	0.947
Baidu.com	0.998	0.992	0.993	0.999	0.994
Amazon.com	0.999	0.999	0.999	0.999	0.994
Yahoo.com	1.000	1.000	0.998	1.000	0.999
Wikipedia.org	1.000	1.000	0.999	0.100	0.999
Qq.com	1.000	1.000	1.000	0.999	0.999
Twitter.com	0.999	0.999	0.999	1.000	0.999
Google.co.in	1.000	1.000	1.000	1.000	1.000

A: Internet Explorer B: Mozilla Firefox
C: Google Chrome D: Safari E: Opera

ことを示した。これらの結果から、W-style は、Web ページデザインに対する閲覧者からのフィードバックを促し、開発者が Web ページ開発をする際に有用となる情報を取得する有効な手段の 1 つであるという結論を得た。

今後の課題として、追加の評価を検討している。今回の評価では、Web ページのデザインを編集する作業を 4 つに絞って行ったが、今後、数多くの編集パターンにおいて

Web ページのデザインが崩れないことを検証する必要があると考える。

参考文献

- [1] Ahuja, A.: Stylebot Social, stylebot.me (online), available from <http://stylebot.me/> (accessed 2015-12-10).
- [2] Alexa Internet, Inc.: Alexa Top 500 Global Sites, Alexa Internet, Inc. (online), available from <http://www.alexa.com/topsites> (accessed 2015-12-15).
- [3] JotForm Inc.: Wishbox: Easiest Feedback Tool, JotForm Inc. (online), available from <http://www.jotform.com/wishbox/> (accessed 2015-12-10).
- [4] Q-Success: Usage Statistics and Market Share of Server-side Programming Languages for Websites, January 2016, Q-Success (online), available from http://w3techs.com/technologies/overview/programming_language/all (accessed 2016-01-12).
- [5] Q-Success: Usage Statistics of Client-side Programming Languages for Websites, January 2016, Q-Success (online), available from http://w3techs.com/technologies/overview/client_side_language/all (accessed 2016-01-12).
- [6] StatCounter: Top 5 Desktop, Tablet & Console Browsers from Dec 2014 to Dec 2015 — StatCounter Global Stats, StatCounter (online), available from <http://gs.statcounter.com/#browser-ww-monthly-201412-201512-bar> (accessed 2016-01-12).
- [7] W3C: Document Object Model (DOM) Level 3 Core Specification, W3C (online), available from <http://www.w3.org/TR/DOM-Level-3-Core/> (accessed 2016-02-04).
- [8] W3C: HTML5, W3C (online), available from <http://www.w3.org/TR/html5/> (accessed 2015-12-09).
- [9] Wang, Z., Bovik, A. C., Sheikh, H. R. and Simoncelli, E. P.: Image Quality Assessment: From Error Visibility to Structural Similarity, *IEEE Transactions on Image Processing*, Vol. 13, No. 4, pp. 600–612 (2004).
- [10] 木浦幹雄, 大平雅雄, 上野秀剛, 松本健一: 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および評価, 情報処理学会論文誌, Vol. 51, No. 1, pp. 204–215 (2010).
- [11] 小箱雅彦: 電子化文書の画像圧縮ガイドライン, 月刊 IM, Vol. 50, No. 5, pp. 21–24, 公益社団法人 日本文書情報マネジメント協会 (2011).
- [12] 若松祐樹, 早川智一, 疋田輝雄: Web ページのレイアウト調整システムの提案, 第 13 回情報科学技術フォーラム講演論文集 第 4 分冊, pp. 239–244 (2014).