

# WePatch 2 : Web フォームにおける入力ミスを利用した半自動 BADUI 改善システム

田島一樹<sup>†1</sup> 中村聡史<sup>†1</sup>

**概要** : Web 上には間違えた入力をしやすい登録フォームや誤りがある情報提示といったような、ユーザの失敗や誤解を誘発するようなユーザインターフェース (BADUI) が多く存在している。また、その改善の手間などから Web サイトの BADUI は放置されることが珍しくなく、日々ユーザを困らせている。我々はこれまでの研究で、BADUI における適切な操作のための手がかりの提示や入力の自動修正といった、改善機能の付与と共有をエンドユーザに可能とさせるシステムを実装した。しかし、過去に提案した手法は改善に手間がかかることや、不適切な改善がなされる可能性があるという問題があった。そこで本稿では、システムがユーザの失敗行動から適切な改善機能の推薦を行う半自動的な BADUI 改善手法を提案する。また、提案手法を Web フォームにおいて実現するシステムを実装し、使用実験を行った結果、エラー率の減少やユーザビリティの向上が可能であることを明らかにした。

**キーワード** : Web フォーム, BADUI, ユーザビリティ, ログ分析

## 1. はじめに

Web 技術の進歩と利用するユーザ数の増加によって、便利な Web サービスは日々増えつつある。例えば、オンラインショッピングやシェアリングサービス、イベントへの参加登録のように、ユーザの様々なニーズに沿った Web サービスが作成されており、またその数は膨大なものとなっている。ここで、Web サービス運営者にとって、ユーザに競合サービスではなく自身の運営するサービスを利用し続けてもらいつつ、そのサービスからの離脱率を下げるためにユーザビリティは重要である[1]。

しかし、開発時にユーザビリティの適切な評価を行わなかったために、ユーザの失敗や誤解を招くようなユーザインターフェース (BADUI) が出来てしまうことは少なくない。また、開発における予算が少なかったり開発期間が短かったりするなど開発にかけることが可能なリソースが不十分な場合や、開発時に想定していなかった問題の発生により、結果的に BADUI となってしまうことも珍しくない。こうした BADUI はサービスを公開した後であっても、サービスの再評価やユーザからのフィードバックによって原因を特定し、改善するべきであるが、サービス運営者の予算やリソース不足、ユーザビリティを重視しない運営者の存在などにより、改善されずに放置されることも珍しくない。

我々はこれまでの研究[2]において、Web 上に放置された BADUI の問題を解消し、ユーザの Web 上での失敗やストレスを減らすため、操作のための手がかりの提示や入力内容を適切な形に自動修正するといった改善機能をユーザの手で付与可能とするシステムである WePatch を提案および実装してきた。このシステムによって、Web フォームにおける入力内容の自動修正、入力例の提示、付箋型アノテーションによる操作上のヒントの提示といった BADUI にお

けるユーザの操作支援を行うことが可能となった。また、WePatch の機能の中でも、Web フォームで多くみられる誤った形式での入力を招くような BADUI (図 1) において、ユーザが誤った形式で入力した文字を自動的に正しい形式に変換する機能が有用であることを明らかにしてきた[3]。

しかし、WePatch を用いてユーザが改善機能を付与するためには、ユーザ自身が BADUI に対して適切な改善機能を選択し、クリックや文字入力を行なった上で共有という手順を踏むため、操作上の手間がかかるものであった。また、WePatch の改善機能の中でも付箋型アノテーションは、そちらへの注意をひいてしまうがために、ミスを誘発する可能性が明らかになっており、意図せずに使いづらくしてしまう問題もあった。

そこで本稿では、従来の WePatch における、改善のための操作の手間や、意図しない UI の改悪が起こる問題を減らすため、ユーザの手による半自動的な BADUI 改善手法を提案する。具体的には、Web フォームにおけるユーザの失敗行動から BADUI を検知し、システム側から改善機能の推薦を行う。また、本稿では提案手法を Web フォームの BADUI を対象として実現する WePatch 2 を実装し、Web フォームにおける BADUI の改善を行う使用実験により、システムの有用性やリミテーションを検証する。

名前(全角):	田中太郎
ふりがな(かたかな):	たなかたろう
NAME(英字大文字):	TANAKATARO

図 1 ふりがなを誤って平仮名で入力してしまう BADUI

<sup>†1</sup> 明治大学  
Meiji University

## 2. 関連研究

### 2.1 Web 拡張によるユーザ支援

Web 拡張を用いてユーザの操作を支援する研究はこれまでに多くなされている。Dong[4]らは、Web サイト運営者のユーザビリティ評価に対する専門知識とリソースの不足を解消するため、Web 拡張を用いて分かりづらいテキストの修正や、ツールチップの付与をユーザのコミュニティーが行うことによって問題を改善可能とするシステムを実現している。Dong らは Web 上のテキストを改善対象としているが、本稿で実現するシステムは入力を間違えやすい Web フォームを改善対象としている点で異なる。Dale[5]らはユーザが Web サイトに付箋型のアノテーションを付与し、他ユーザとの共有を可能とするシステムを実現しているが、こうしたテプラ型のアノテーションは、ユーザを逆に混乱させてしまう可能性があることが我々の過去の研究[3]で示唆されている。Samuel[6]らは Web 上でユーザ同士のリアルタイムなコミュニケーションを実現しており、操作方法を質問及び回答できるが、こうしたサービスではその場、その時間に対話できるユーザが必要である。一方、我々の手法では Web サイト上に機能を残しておくことができるという利点がある。

これらの研究のように放置されている BADUI を、ユーザが改善可能とするツールは多く存在するが、我々はこうしたツールにおける操作の手間や意図しない UI の改悪といった問題をなくすことで、ユーザによるユーザビリティ改善研究に貢献するものである。

### 2.2 ユーザビリティ診断

これまでに Web サイトのユーザビリティの計測や問題点の特定を行うことを目的とした研究は多く行われている。代表的なユーザビリティ評価法の一つとしてヒューリスティック評価法[7]が挙げられる。これは 3~5 人の専門家によってユーザビリティの問題点の評価と改善を繰り返し行うものである。また、FUS (Form Usability Scale) [8]のようにアンケートによって Web フォームのユーザビリティを測定する手法も提案されている。さらに、ユーザの視線ログデータを分析することによって問題点を検知する手法も提案されている。しかし、これらの手法では人手やコストが必要なことから、運営者が実施できない可能性が問題として考えられる。一方、我々の手法では自動で BADUI の検出から改善まで行うため、そうした評価に必要な人手を必要としないというメリットがある。

また、ユーザビリティに対してコストや労力をかけられない運営者のために、自動的に BADUI を検知するような研究が行われている。Grigera ら[9]は、ユーザの多量なイベントデータを分類、フィルタリングすることで、自動で即時的なユーザビリティの問題点の検知を可能とするシステムを実現している。こうしたシステムは開発者にユーザビ

リティの問題点を報告するものが多いが、ユーザビリティに関心がない運営者によって放置されてしまうという問題がある。一方、我々の手法では問題点の自動検知から改善まで行うことが可能である。

### 2.3 ユーザビリティガイドライン

これまでの Web サイトのユーザビリティの知見を集約した、開発者が守るべきガイドラインは多く提案されている。Nielsen ら[10]は UI デザインで守るべき 10 項目を定めており、10 ヒューリスティクスという名前で知られている。このヒューリスティクス評価法は、客観的にユーザビリティの問題点を発見したい場合に用いられることが多い。しかし、こうした確認項目は抽象的であり、Web フォームのユーザビリティを評価する際には、Web サイトの種類に適したガイドラインを用いる必要がある。その 1 つとして、Javier[11]らは Web フォームにおける 20 個のガイドラインを提案しており、手軽に Web フォームのユーザビリティの向上が期待できるものである。また、Mirjam[12]らによってこうしたガイドラインが実際に有用であることも明らかにされている。こうしたガイドラインの有用性が明らかになっているにも関わらず、ユーザビリティを重視しない開発者がガイドラインを参考にしないことで、BADUI のある Web サイトが作成されてしまうことは少なくない。一方で、本稿で提案する WePatch 2 はこうしたユーザビリティの評価から漏れてしまった BADUI におけるユーザの入力形式を修正することが可能である。

## 3. WePatch 2

### 3.1 提案手法

我々は既存のユーザの手による Web 上の BADUI 改善手法をより手間なく、かつ信頼性の高いものにするため、ユーザの失敗および修正行動から、その Web サービスが本来求めている入力形式を推定し、入力補正を行う改善機能の付与を推薦する、半自動的な BADUI 改善手法を提案する。

例えば、ある Web 上の入力ボックスでは「カタカナ」しか受け付けませんが、ユーザがそれに気づかず「ひらがな」で入力して送信ボタンを押したとき、システムは何らかのエラーメッセージを表示する。ここで、「カタカナで入力してください」などの丁寧なエラーメッセージが提示されている場合は、ユーザはそのメッセージに従い、カタカナで入力し直し、送信ボタンを押すであろう。一方、「エラーが発生しました」や「受け付けできない文字が入力されました」などのように、何が誤っているのかわかりにくいメッセージの場合、ユーザは困惑しながらもこれまでの経験などを活かして色々と入力内容を修正する。このときユーザは、「プログラムやデータベースが本来求めている形式はカタカナなのではないか？」と推定することで、正しい入力形式（ここではカタカナ）へとたどり着くことができるかもしれない。また、正しくカタカナで入力した後、送信ボ

タンを押すと、エラーメッセージは表示されず、サービスへの登録などが完了する(図2)。我々の目的はこうしたWebフォームにおける試行錯誤の失敗と成功の入力内容の差から、ユーザには見えないプログラムやデータベースが求める正しい入力フォーマットを推定し、推定した結果をもとに、そのWebフォームに対して適切な変換フィルタ(この場合は、ひらがな→カタカナ変換フィルタ)を推薦することである。提案手法におけるBADUI検知と改善機能付与までのプロセスを図3に示す。

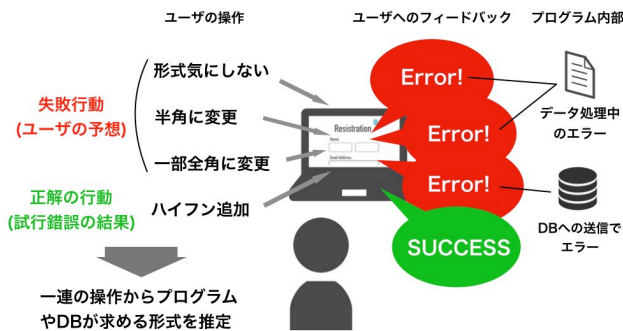


図2 ユーザのトライアンドエラー

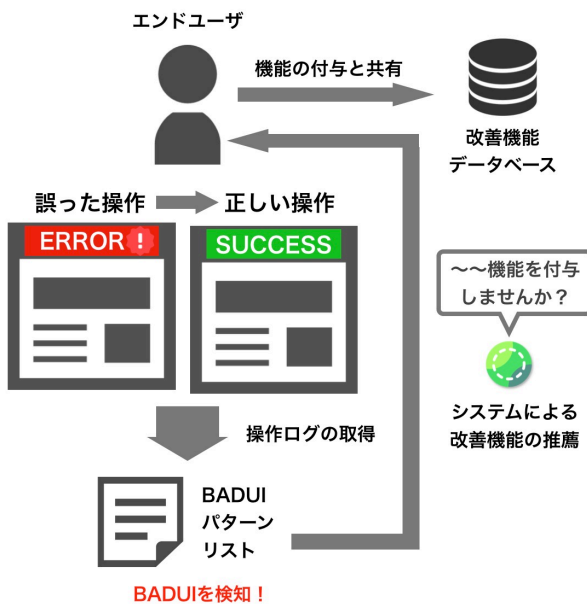


図3 BADUI検知と改善プロセス

### 3.2 Web上のテキスト入力に関するBADUI

本稿では、Web上のテキスト入力ボックスについて、そのBADUIの自動検知を行う。一般にWeb上のテキスト入力ボックスに関するBADUIは、入力データを処理するプログラム(日付の入力としてYYYY-MM-DDとなっているものを、ハイフン区切りでデータ取得するなど)と、データを格納するデータベースのセルのフォーマット(整数、実数、半角英数字、ハイフン区切りのDATEフォーマット、

文字数制限など)に依存していることが多い。

例えば、図4に示すテキスト入力フォームは、口座番号を半角数字7文字で入力する必要があり、6桁の場合は先頭に0を、5桁の場合は先頭に00を追加する必要がある上、受取人名は、半角のカタカナと丸括弧のみ入力可能である。このようなBADUIの場合、ユーザはかなり混乱してしまうことになる。また図5では、「\*\*番地・部屋名に受付できない文字が入力されました。ご確認の上、訂正してください。」とあるが、どこに問題があるのかわからないために、ユーザをかなり混乱させてしまうものである。なおこのフォームの場合は、全角文字しか許容せず、半角の英数字、半角の空白文字がエラーの原因となっていた。

図4や5および、これまで収集された膨大なBADUIから、テキスト入力ボックスについては、下記の組み合わせの制約があり、結果的にBADUIになっているといえる。

- **文字種の制約**：ひらがな、カタカナ、数字、漢数字、アルファベットのいずれか、またはその組み合わせ
- **記号に関する制約**：ハイフン必須/不要、スラッシュ必須/不要、使用可能な記号の制約(例：アンダーバーのみ、ハイフンのみ、\$#!?のみ)など
- **半角・全角**：半角文字のみ、全角文字のみ、両方を許容するなど
- **数値データに関する制約**：整数、実数、桁数に応じて先頭に0を埋める(例：8月を08月と入力)など
- **文字数制限**：8文字ちょうど、10文字以内、140字以内、1000文字以内など

なお上記の組み合わせによるBADUIは、その入力形式についての説明が欠落していたり、説明がわかりにくかつ

図4 BADUIの例

転送ご依頼の場合は、配達時間帯のご指定はできませんので、ご了承ください。

#### 転送先情報の入力

\*\*番地・部屋名に受付できない文字が入力されました。ご確認の上、訂正してください。

図5 BADUIの例

たり、間違っていたり（ひらがなで入力しなければならないのにカタカナと書いてあるなど）、エラーメッセージがわかりにくかったり、初期値として入力されているものが間違っている（整数で入力しなければならないのに実数値が入っているなど）といったことがそのわかりにくさに拍車をかけている。

### 3.3 機能

BADUI を検知した後の機能として、WePatch 2 では以下の3つの機能を実装した。

- **変換フィルタの推薦**：ユーザの誤り訂正操作からBADUI を検知するとボタンと吹き出しが表示される。ユーザはボタンを押すことでフィルタを適用可能である。また、誤ってクリックしてしまった場合であってもフィルタを取り消すことが可能である。ユーザによる誤った形式での入力と、入力修正後にフィルタが推薦される様子を図6, 7に示す。
- **入力の自動修正**：フィルタが付与された入力フォームでは、ユーザが入力した文字に対しリアルタイムに修正が行われる。修正時は文字が点滅するフィードバックがあり、ユーザは入力しながら文字の修正が行われていることを把握可能である。
- **フィルタの評価**：自身の文字形式が修正された場合に、フィルタに対して高評価または低評価をすることが可能となっている。これにより、他のユーザによる投票数からフィルタの信頼性を把握可能である。図8はフィルタの評価ボタンが提示されている様子である。

### 3.4 実装

WePatch 2\*は Chrome 拡張として JavaScript と PHP を用いて実装した。また、Web フォームに付与された変換フィルタの情報の共有は PHP と MySQL の連携によって可能とした。変換フィルタは Google Chrome でこの拡張機能をダウンロードしたユーザ間で反映される。JavaScript は主に入力ボックスでの入力中に文字のカタカナや半角といった形式情報の取得とウェブストレージへの一時保存、エラー前後の形式の比較と差異の抽出、変換フィルタの推薦、文字修正といった処理の実行を行うために用いた。

WePatch 2 は Web フォームにおいて元々ユーザがいたページに戻った場合をエラーと定義した。また、エラー前に入力された文字形式情報はウェブストレージに格納しておくことで、エラー後の文字形式と比較することを可能とした。入力形式の取得は、入力ボックスへのフォーカス中にエンターキーを押すことでボタンが提示されずに送信されることを防ぐため、入力形式はユーザの入力中にリアルタイムにチェックを行なった。

また、PHP は Ajax の非同期通信による値の受け渡しと、MySQL に対する付与された時刻、URL、改善されたフォー

ムの ID、付与されたフィルタの種類、フィルタの評価情報といった改善情報に関する値の送信と取得のために用いた。

図 6 誤った形式での入力

図 7 フィルタの推薦の様子

図 8 変換フィルタに対する評価の提示

## 4. 評価実験

WePatch 2 により Web フォームが改善されるかどうかをユーザベースの利用実験から評価する。ここでまず、Web フォームに関する BADUI の収集と、BADUI を含む Web フォームの作成を行う。また、WePatch 2 を用いた改善タスクに取り組んでもらい、実験協力者全員のタスク終了後、付与された変換フィルタ情報、エラー率、また実験後アンケートの結果を用いて有用性を検証する。

### 4.1 Web フォームの BADUI 収集と BADUI ページ作成

BADUI 収集には BADUI に関する大学の講義を受講済みの 18~20 歳の大学生 11 名(男性 8 名女性 3 名)を募集し、自由に Web 上の登録フォームを検索してもらい、そのなか

\* <http://tajima.nkmr.io/wepatch/>



らユーザが開発者の意図とは異なる形式で入力してしまうような BADUI を集めるよう依頼した。収集した BADUI の中で、登録フォームにまつわるものは 16 種類あり、入力形式を間違えてしまう原因となる Web フォームの要素で分類した結果、入力形式の手がかり（ひらがな、半角等を指定する文字列）に関するものが 7 件、入力ボックスの隣にある入力項目のタイトル（お名前、住所等）に関するものが 4 件、入力必須項目の手がかりに関するものが 3 件、入力例に関するものが 1 件、入力ボックス数に関するものが 1 件であった。以上の収集結果より、表 1 に示す BADUI を選定した。

表 1 実験に用いた BADUI

名前	ひらがな入力必須だが、ラベルと例が「ヒラガナ」と「タナカタロウ」のようにカタカナで表記されている
	カタカナ入力必須だが、ラベルと例が「かたかな」と「たなかたろう」のようにひらがなで表記されている
	英字大文字での入力が必須で、ラベルも英字大文字であるが入力例が英字小文字で表記されている
	英字入力を指定されているフォームが、全角英字で入力必須である
年齢	一般的に数値の入力は半角だが、全角数字での入力が必須である
生年月日	年月日を区切るスラッシュが必要だが、スラッシュ入力に関する手がかりがない
住所	郵便番号が半角入力必須であるにも関わらず、住所は全角入力が必須である
	電話番号ではハイフン入力が必要でないが、郵便番号ではハイフンの入力が必須である
	ひらがな入力必須だが、入力例がカタカナで表記されている
電話番号	ハイフンを含めた入力が必須だが、手がかりがない。

なお、過去の実験[3]で作成した Web フォームはあまりに入力項目が多く、その負荷も大きくなってしまったため、本稿で用意する Web フォームの入力内容は 1 つのページにつき 2~3 項目とした。また、表 1 を参考に、ページごとに入力項目やタイトル、入力形式の手がかり、入力例を用意した。また、背景画像やフォント、配色といったデザインを 1 ページごとに異なるようにした。具体的な入力項目は、名前に関する情報（漢字、ふりがな）、年齢に関する情報（年齢、生年月日）、住所に関する情報（郵便番号、住所）、

連絡先に関する情報（電話番号、メールアドレス）である。図 9 は、実験用の Web フォームの例である。この Web フォームは、入力形式が合っていれば次のページに進み、誤って入力された場合にフォームの隣にエラーを提示するものである。なお、実験用の Web フォームとしては、全部で 50 ページを用意し、その内の 10 ページに表 1 の BADUI が含まれるようにした。ここで、BADUI でないページを実験用ページとして用意した理由は、ユーザの使いづらい UI への慣れや警戒心を薄め、また、改善されやすいページやエラー率がページごとに偏ってしまうことを防ぐためである。



図 9 実験用 Web フォーム

## 4.2 BADUI 改善タスク

BADUI 改善タスクの準備として BADUI 収集を依頼した実験協力者とは別の 13 名を募集した。実験協力者は、21~23（平均 22.5）歳の男性 11 名、女性 2 名であり、PC の使用経験年数は平均 6.4 年、タッチタイプを習得している実験協力者は 5 名、ある程度タッチタイプが出来る実験協力者は 7 名、キーボードを見ながらでしか入力できない実験協力者は 1 名であった。

実験協力者には普段から利用している PC に WePatch 2 を拡張機能に追加してもらい、使い方を教えた後、推薦された変換フィルタが適切であると感じたら適用してもらい、不適切なものだと感じたら付与しないように指示した。適用されたフィルタに関しては、本研究では他の実験者に共有された状態で提示される。また、入力作業については、入力するプロフィールはこちらが用意したダミープロフィール\*を用いて入力するよう指示した。さらに、ユーザはページ上で提示されている情報を参考にして、適切な形式で入力を行なってもらうようにした。

練習用の入力フォームでプロフィールを 3 回入力してもらった後、BADUI を含めた実験用入力フォームで BADUI 改善タスクを行なった。なお、実験は連続的に実施されるため、徐々に BADUI 改善フィルタが適用されていくことになる。そのため、後半の実験協力者ほど BADUI の問題が改善されていると期待される。

\* <https://hoge hoge.tk/personal/>

また、実験後は Web フォーム全体のユーザビリティと心的負荷を測るための 2 種類のアンケートを実施した。入力フォームのユーザビリティの測定には FUS (Form Usability Scale) [8]を用い、ユーザの心的負荷を測定には日本語版 NASA-TLX (NASA Task Load Index) [13]を参考にし、Web フォームに向けたものに変更して用いた。アンケートの質問項目を以下に示す。

#### FUS

- 実験のフォームは長さが適切と感じた
- 素早く実験のフォームを入力出来たと感じた
- 質問項目の順序を適切と感じた
- どういった情報を入力するべきかが常に理解出来た
- どういった制限に従うべきかを全体的に理解できた
- エラーが発生した場合、その問題を解決する方法が明らかなエラーメッセージが表示された
- 実験用入力フォームに満足している。

各質問項目に対する回答は、「全くそうは思わない」、「あまりそうは思わない」、「どちらでもない」、「まあまあそう思う」、「とてもそう思う」、の 5 つの中から一つに選択する、5 段階のリッカード尺度を用いる。

#### NASA-TLX

- タスクを実行中に、フォームを見る、フォームの指示を確認する、考える等どれくらいの知覚的活動が必要だったと感じましたか
- タスクを実行中に、入力する、クリック等、どれくらいの身体的活動が必要だったと感じましたか
- タスクを実行するにあたって、この課題の頻度または速度から感じた時間的圧力はどの程度だったと思いますか
- 与えられた課題の達成にどの程度努力したと思いますか
- 課題目標について、あなたはどの程度成功したと思いますか
- 作業中に、いろいろ、不安、落胆、ストレス、悩み等をどの程度感じましたか

各質問は小～大、少ない～多い、悪い～良い、を 0～11 の 11 段階のリッカード尺度を用いる。

## 5. 結果

実験終了後、実験用 Web フォームにおけるユーザのエラー率を確認したところ、BADUI を含んだ 10 のページ中、4 ページについては 13 名の実験協力者のいずれもが失敗しなかった。なお、1 名以上が失敗した 6 ページの BADUI は、表 1 における名前の 1, 2, 3 列目、年齢の 1 列目、住所の 1, 3 列目に該当している。

また、1 名以上が失敗した BADUI のページについては、システムにより BADUI として検知されていた。なお、適用されたフィルタを確認すると、6 ページ中 3 ページは正

解の変換フィルタであったが、残りの 3 ページは不正解の変換フィルタが適用されていた。なお、不正解のフィルタが付与されたのは表 1 における名前の 1 列目、住所の 1, 3 列目の BADUI であった。

次に、フィルタが適用されるごとにエラー率が下がるかどうかを確認するため、実験協力者の実験を行なった順番にアンケート結果のプロットを行う。なお、エラー率に関しては、各実験協力者がエラーを出力したかどうか（エラーがなければ 0、出力されれば 1）をページごとに平均し、割合を算出する。図 10, 11 の縦軸は平均エラー率であり、横軸は実験協力者が全体の中で何番目に実験を行なったかを表す。実験順番に対応したエラー率の減少傾向を示すため、目的変数を平均エラー率、説明変数を実験協力者の順番として単回帰分析を行なった。結果として、BADUI が含まれる 10 ページにおける平均エラー率を用いた近似式では、二つの要因は弱い負の相関関係（決定係数=0.3534）にあり、全ページにおける平均エラー率を用いた近似式でも、二つの要因は弱い負の相関関係（決定係数=0.2904）にあることが明らかになった。よって、フィルタが付与されるごとにエラー率を減少させることが可能であることが明らかになった。

フィルタが付与されるごとにユーザビリティの向上やタスクによる心的負担の軽減度合いを確認するため、実験協力者の実験を行なった順番にアンケート結果のプロットを

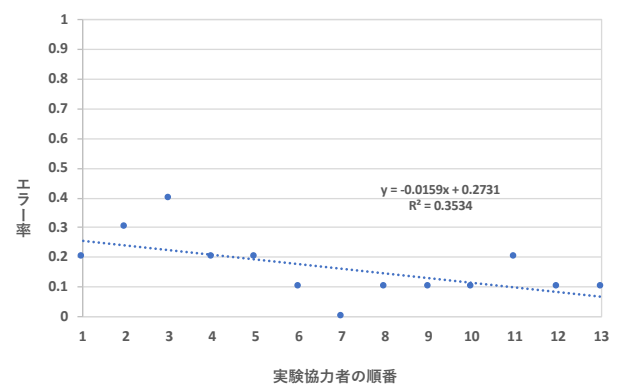


図 10 エラー率の推移 (BADUI ページのみ)

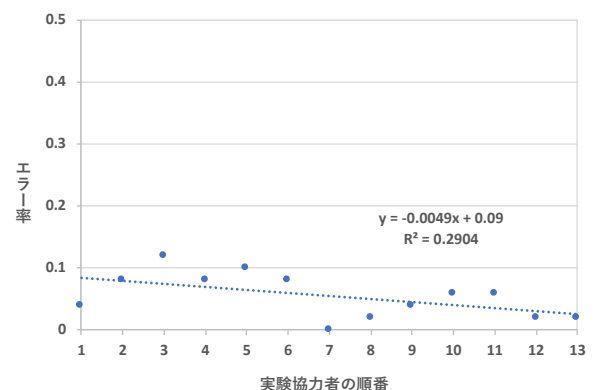


図 11 エラー率の推移 (全ページ)

行う。ここで、プロットを行うデータはアンケートの各質問項目に回答された全ての値を平均したものである。

まず、FUS への回答の分析結果を図 12 に示す。図の縦軸は全質問項目への評価値を平均した値であり、横軸は実験協力者が全体の中で何番目に実験を行なったかを表す。実験順番に対応したユーザビリティの上昇傾向を示すため、目的変数を評価平均値、説明変数を実験協力者の順番として単回帰分析を行なった。結果として、近似式（決定係数=0.514）より、2つの要因は弱い正の相関関係にあった。

次に、NASA-TLX への回答の分析結果を図 10 に示す。図の縦軸と横軸は図 13 と同様な意味を表す。実験順番に対応した心的負荷の減少傾向を示すため、目的変数を評価平均値、説明変数を実験協力者の順番として単回帰分析を行なった。結果として、近似式（決定係数=0.1383）より、2つの要因に相関関係は示されなかった。

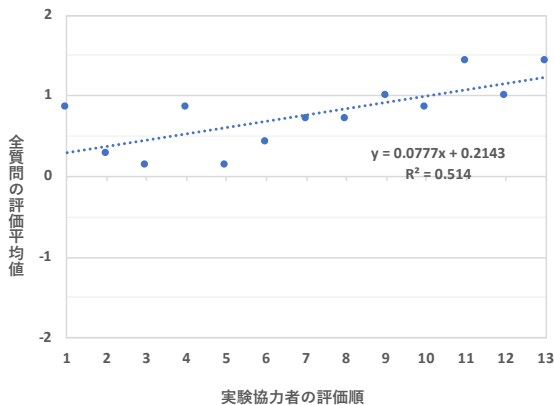


図 12 評価平均値の推移 (FUS)

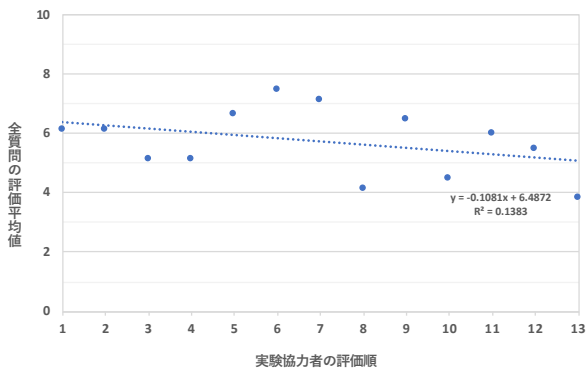


図 13 評価平均値の推移 (NASA-TLX)

## 6. 考察

実験の結果より、Web フォームの BADUI におけるユーザの失敗行動の検知、また半自動的な変換フィルタによる改善を行うことで、エラー率の減少やユーザビリティの向上が可能であることを明らかにした。そのため、WePatch 2 を利用するユーザ間で、Web フォームを利用される度に

BADUI の入力ボックスにフィルタが適用されていき、ユーザビリティが向上すると考えられる。

しかし、不正解の変換フィルタが付与されたページもあり、こうした問題に対処するには BADUI 検知アルゴリズムの改良が必要であると考えられる。具体的には、現在のアルゴリズムではユーザがエラー前に入力した最後の文字の形式と、エラー後の最初に入力した文字の形式を比較する形となっているため、どうしてもそのユーザの試行錯誤を判断することができない。そこでリアルタイムなチェックは行いつつも、最終的に入力完了後の形式も考慮した推薦を可能とすることで精度を高める予定である。また、1人のユーザの失敗からの修正のみから変換フィルタをシステム側で選定したため、そのユーザの修正が適切でない場合に不正解の変換フィルタが推薦されてしまう可能性が考えられる。そこで、複数ユーザの修正行動から多数決で変換フィルタの選定を行うよう実装することで、より信頼性の高い変換フィルタの推薦が可能になると考えられる。また、BADUI 検知アルゴリズムを改良し、どの程度 BADUI 検知が可能であるかの精度を検証する必要があると考えられる。

今回、全実験協力者が一度もミスをしなかったページは 4 つあったが、これは 1 ページにおける入力項目が少ないことが原因であると考えられる。そこで今後は、BADUI ページの難易度を高めることによって、BADUI 検知が可能かどうか検証する予定である。

また、今回用意した手法は、3.2 節で触れた文字種の制約や記号に関する制約、半角・全角に関する制約や、数値データに関する制約、文字数制限の全部をカバーできているわけではなかった。そこで今後は、こうした様々な入力について考慮しつつ、BADUI を改善していく予定である。

ここで、今回提案する手法は、多くのユーザに対して入力を監視されているのではという不安を抱かせるものとなっている。実際にはユーザが変換フィルタ付与ボタンを押してはじめて、そのフィルタに関する情報のみが送られるようになっている。しかし、なんらかの情報を送っているというのは、ユーザに対して積極的に使おうと思わせるには弱いものであるといえるため、ユーザに対して不安感を抱かせないものにするようシステムを改良する予定である。

## 7. まとめと今後の展開

本稿では、従来のユーザの手による BADUI 改善手法における手間と意図しない UI の改悪を減らすため、ユーザの試行錯誤の情報を用いて BADUI の検知を自動的に行い、ユーザに改善機能の付与を推薦する半自動的な BADUI 改善手法を提案した。また、Web フォームにおいて BADUI を検知し、変換フィルタの推薦を行う WePatch 2 を実装した。評価実験により、Web フォームにおけるエラー率を減少可能であること、また、ユーザビリティを向上させることが可能であることを明らかにした。

今後は、Web フォームにおける BADUI 検知精度の検証実験、また、半自動的な BADUI 改善手法を変換フィルタ以外の機能で実現を行う予定である。さらに、ユーザが BADUI 改善のモチベーションを維持するため、どの程度改善機能によって他のユーザが支援されたか、といった情報をグラフなどによって可視化を行う機能を実装する予定である。

**謝辞** 本稿の一部は JST ACCEL (グラント番号 JPMJAC11602) の助成を受けたものです。

## 参考文献

- [1] Wroblewski, L.. Web Form Design: filling in the blanks. Rosenfeld Media, 2008, 226p.
- [2] Tajima, K. and Nakamura, S.. WePatch: A System Enabling Users to Improve Bad User Interfaces on the Web. the 29th Australian Conference. 2017, p. 448-451.
- [3] 田島一樹, 中村聡史. WePatch : ユーザの手による Web 上の BADUI 改善システムの評価. 情報処理学会研究報告ヒューマンコンピュータインタラクション (HCI) . vol.174, no. 16, p.2188-8760.
- [4] Dong, T., Ackerman, M. S., Newman, M.W. and Paruthi, G.. Socialoverlays:Collectively making websites more usable. INTERACT 2013. 2013, vol. 8120, p. 280-297.
- [5] Reed, D. and John, S.. Web annotator. SIGCSE 2003, ACM, vol. 35, p. 386-390.
- [6] Sergio, T.S. and Ackerman, I. I. Enabling communication between users surfing the same web page. 2003.
- [7] Nielsen, J.. Heuristic Evaluation. In Usability Inspection Methods. John Wiley & Sons, Inc., 1994.
- [8] Aeberhard, A.. FUS - Form Usability Scale. Development of a Usability Measuring Tool for Online Forms. Unpublished master's thesis. Unpublished master's thesis. 2011.
- [9] Grigera, J., Garrido, A., Rivero, J. M. and Rossi, G.. Automatic detection of usability smells in web applications. International Journal of Human-Computer Studies. 2017, vol.97, p.129-148.
- [10] "10 Usability Heuristics for User Interface Design". <https://www.nngroup.com/articles/ten-usability-heuristics/>, (参照 2018-07-25).
- [11] Bargas-Avila, J, A., Brenzikofer, O., Roth, S., Tuch, A, N., Orsini, S. and Opwis, K.. Simple but Crucial User Interfaces in the World Wide Web: Introducing 20 Guidelines for Usable Web Form Design. 2010.
- [12] Seckler, M., Heinz, S., Bargas-Avila, J, A., Opwis, K., Tuch, Alexandre, N.. Designing usable web forms. 2014. Proceedings of the 32nd annual ACM conference on Human factors in computing systems, pp.1275-1284.1
- [13] 三岳晋司, 神代雅晴. メンタルワークロードの主観的評価法—NASA-TLX と SWAT の紹介および簡便法の提案—. 人間工学, 1993, vol. 29, no. 6, p. 399-408.