

2018 年度 修士学位請求論文

ユーザの手による Web フォームの
BADUI 改善手法

明治大学大学院先端数理科学研究科

先端メディアサイエンス専攻

田島 一樹

Master's Thesis

A Method Enables Users to Improve Bad User
Interfaces on Web Forms

Frontier Media Science Program,

Graduate School of Advanced Mathematical Sciences,

Meiji University

Kazuki Tajima

概要

Web 上には多くのユーザの失敗や悩みを誘発するようなユーザインタフェース (BADUI) が多く存在している。その中でも、特に Web フォームでは、ユーザの誤った入力を誘発してしまうものや、入力のための手がかりが欠如している BADUI が存在し、日々利用する人々を困らせている。ここで、Web フォームは、UI と内部のプログラム及びデータベースが求める文字形式の間にギャップが存在する場合や、UI に内部が求める文字形式が表記されていない場合に BADUI となることが多い。こうした原因と Web フォーム内に配置されている入力ボックス数から、誤った文字形式の入力をしてしまう BADUI は 8 種類に分類できる。

過去の研究において、BADUI の問題を解決するため、開発時に守るべき Web フォームのガイドラインやヒューリスティクス評価法のようなユーザビリティの問題点を見つけ出すための研究が盛んに行われている。また、サービス公開後に BADUI の自動検知と開発者への報告を可能とするシステムが開発されている。さらに、ユーザからのフィードバックを参考にすることで BADUI を改善するといった解決策が考えられる。しかし、こうした手法が有用であるにも関わらず、サービス運営者のリソース不足やユーザビリティへの意識の低さといった理由により、Web フォームの BADUI が放置されることは珍しくない。

我々はこれまでの研究で、放置された Web フォームの BADUI の改善を目的として、エンドユーザが Web フォームの BADUI を改善可能とする手法を提案してきた。また、手法を実現するため、付箋型アノテーションや入力例付与による手がかり提示機能、変換フィルタによる入力の自動修正機能の付与を可能とするシステムである WePatch を実装した。しかし、WePatch の問題点として、改善機能が Web フォームにおけるユーザの負担を実際に減らすかどうかは明らかでないこと、また、改善機能を付与するための操作がユーザにとって手間であることの 2 つが挙げられる。

そこで、第 1 に、BADUI に改善機能が付与された後、実際に負担が減るかどうかを検証するため、BADUI の Web フォームにおける入力タスクを実施した。また、その過程でユーザの操作時間やエラー数といった定量的な値を取得し、改善前と後の値の比較を行なった。結果として、入力の自動修正機能はユーザの負担を減らすことが可能であり、従来の研究で有用であると考えられていた付箋型のアノテーションによる手がかり提示は、ユーザの負担を増やしてしまうことが明らかになった。

第 2 に、WePatch の改善機能の付与の手間を減らすことを目的として、Web 上におけるユーザの失敗操作と正解操作を利用することで、BADUI を検知し、ユーザに対して適切な改善機能の推薦を行う、半自動的な BADUI 改善手法を提案した。また、提案手法を変換フィルタにおいて実現したシステムである WePatch 2 を開発した。この WePatch 2 の評価実験として、BADUI を含んだ Web フォームにおける入力タスクと BADUI 改善タスクを行なってもらうユーザベース実験を行なった。その結果として、ユーザによって変換フィルタ

が付与されることにより、Web フォームのユーザビリティが向上することが明らかになった。しかし、この実験では WePatch 2 の改善機能の推薦アルゴリズムや、用意した実験用 Web フォームの設計に課題があることが明らかになった。また、WePatch 2 を BADUI 検知精度の観点から評価していないという問題があった。

そこで、追加の評価実験として、WePatch 2 の推薦アルゴリズムを改良した WePatch 2+ の実装と実験用 Web フォームを改良した上で、ユーザベース実験同様に、BADUI を含んだ Web フォームにおける入力タスクと BADUI 改善タスクを実験協力者に依頼した。結果として、WePatch 2+ はページ内で 5 人のユーザが入力することで 8 割、11 人で 9 割の改善対象の BADUI が改善可能であることが明らかになった。また、半自動 BADUI 改善手法と、その手法における変換フィルタの推薦フェーズを省いた自動 BADUI 改善手法の適合率を比較した結果、半自動 BADUI 改善手法の方が 1 割以上高く、半自動改善手法の方が自動改善手法よりも有用であることが明らかになった。

今後の本研究の応用としては、半自動 BADUI 改善手法により、変換フィルタが付与された BADUI ページの情報を収集することで BADUI データセットを構築し、機械学習を用いて BADUI ページの自動推定や、改善機能の自動付与、開発者への BADUI の報告が可能になると考えられる。

Abstract

Many user interfaces on the Web are confusing for users to operate or make users more prone to mistakes, and we call such user interfaces BAD User Interfaces (BADUIs). In particular, many user interfaces on the Web form annoy users by making them more prone to input incorrectly or lacking clues for input. Now, Web forms often get into BADUI when it has a gap between the UI and the internal programs, and the database requires the character format. We classify BADUIs into eight types based on the causes of the problems and the number of input boxes.

To solving BADUI problems, previous work has proposed many methods which can finding BADUI problems such as heuristics evaluation method and guidelines which developers should follow. In addition, developers should take steps on BADUI by using the system that can automatically detect BADUI or by reflecting on feedback form users after the service release. Although these methods can be done, developers often ignore such user interfaces problems because service operators lack resources or awareness regarding usability for improving BADUIs.

Therefore, to solve these problems, we had proposed a method that enables users to improve BADUIs on the Web. Furthermore, to realize its method, we implemented WePatch which is a system that enables users to attach improvement functions virtually for automatically correcting inputted text (auto-conversion filter), adding input examples (Displaying example), and adding annotations (Web advisory sticker). However, we had not verified its usefulness and had a problem that it takes users time and effort to improve BADUIs.

Then, to verify usefulness of WePatch, we firstly conducted an input task on Web forms including BADUIs after improvement functions were added to them. We obtained data such as a number of error and operation time during the experiment and compared the usability of BADUIs before and after being qualitatively improved by WePatch. The result confirmed that the “auto-conversion filter” in input forms was useful for reducing the burden imposed on the user, but the annotation proposed in the past was proved to be useless because it increased the burden on the user.

Secondly, to reduce user’s time and effort in adding improving functions, we propose a semi-automatic improving method for BADUIs that enables system to recommend correct conversion filter to users by users' trial and error on Web form. We also implemented WePatch 2 that realizes the proposed method on conversion filter. We conducted a user-based experiment in which the participants worked on input tasks and a BADUIs improvement task in Web forms including BADUIs. The experimental result revealed that usability on Web

forms increased when conversion filter was added by the users. However, we found some problems in recommendation algorithm of WePatch 2, and the experimental design, and we did not evaluate WePatch 2 from the aspect of BADUI detection precision.

Therefore, after improving the recommendation algorithm and the experimental design, we conducted another user-based experiment in which the participants were asked to do the same input tasks and BADUIs improvement task as the previous experiment. The result showed that WePatch 2+ could improve 80% of BADUIs when it was operated by five users and 90% of BADUIs when it was operated by eleven users. In addition, we compared semi-automatic improving method and automatic improving method that skip the process of filter recommendation phase from the aspect of BADUI detection precision. As a result, it was revealed that the semi-automatic improving method is more useful than the automatic improving method because precision of semi-automatic method is 10% higher than precision of automatic method.

As a future work, we plan to make BADUI dataset from data of BADUI pages with conversion filter by semi-automatic improving method. In addition, we plan to realize an automatic detection of BADUIs by machine learning, an automatic improvement function, and automatic report of BADUIs to the developers.

目次

第1章	はじめに	1
1.1.	Web フォーム	1
1.2.	BADUI.....	1
1.2.1.	Web フォームの BADUI.....	1
1.2.2.	BADUI の対策とその問題点.....	2
1.3.	本稿の目的	3
第2章	Web フォームの BADUI とその分析	4
第3章	関連研究	8
3.1.	UI デザインガイドライン.....	8
3.2.	人手によるユーザビリティ評価	8
3.3.	自動 UI 分析.....	9
3.4.	ユーザの操作支援	9
第4章	WePatch の評価実験	11
4.1.	WePatch.....	11
4.2.	実験準備.....	12
4.3.	実験手順.....	14
4.4.	結果.....	17
4.5.	考察.....	20
第5章	WePatch 2	22
5.1.	提案手法.....	22
5.2.	Web フォームにおける入力形式の制約	23
5.3.	実装.....	25
第6章	ユーザベース実験	26
6.1.	実験準備.....	26
6.2.	実験手順.....	28
6.3.	結果.....	29
6.4.	考察.....	32
第7章	BADUI 検知精度の評価実験	34
7.1.	WePatch 2+	34
7.2.	実験準備.....	35
7.3.	実験手順.....	37
7.4.	評価指標.....	37
7.5.	結果.....	38

7.6.	考察.....	42
第8章	おわりに.....	44
8.1.	まとめ.....	44
8.2.	リミテーション.....	44
8.3.	今後の応用先.....	44

第1章 はじめに

1.1. Web フォーム

インターネット技術の進歩により、現在の Web サービスではユーザの操作に合わせた動的な表現や幅広いデザイン等が実現可能となった。開発者はこうした技術を活用することで、ユーザの様々な需要を満たす Web サービスを開発し、日々ユーザに恩恵をもたらしている。例えば、過去においては外出が必要であった学会や音楽のライブといったイベントへの参加申し込みや、洋服や食材の購入といった買い物が、現在は Web サービスを通して自宅にいながら気軽に行うことが可能である。

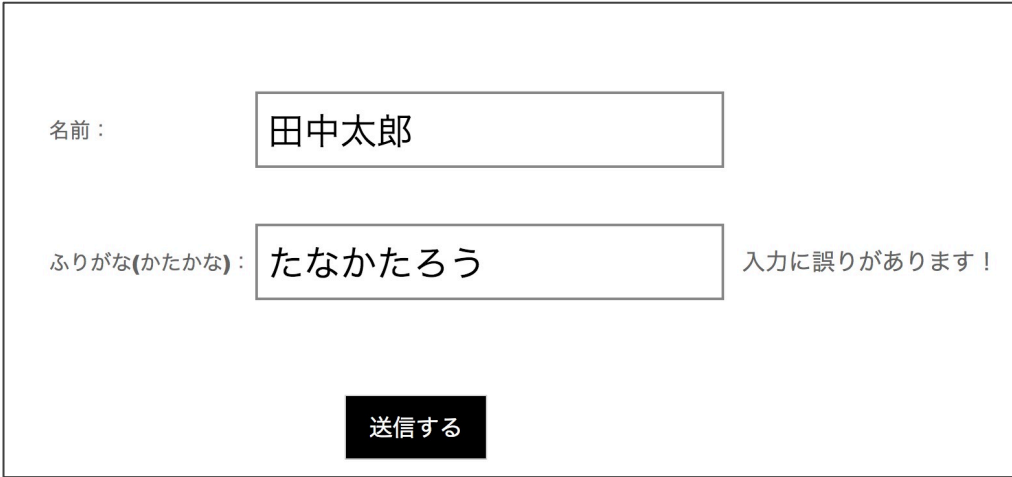
ここで Web サービスにおいて、ユーザがサービスへの登録、問い合わせや感想等のフィードバックの収集、商品の購入、イベントや学会等への参加登録を行うための操作を求めるページが存在する。これらのページは Web フォームと呼ばれ、ユーザが個人情報や任意のコメントといった文字列を入力するためのテキストボックスや、選択式の質問に対する回答するラジオボタン、チェックボックスといった要素によって構成される。運営者にとって、ユーザが Web フォームを用いて送信する情報は、運営するサービスの今後の方向性の決定や、発展の上で重要なものである。一方、ユーザにとって、Web フォームがそのサービスとの最初の接点となることは多い[1]。そのため、Web フォームはユーザが容易に目的を達成するための使いやすさ（ユーザビリティ）が必要である[2]。実際、Nielsen ら[3]は Web の成功の大半はユーザビリティに左右されると述べている。

1.2. BADUI

1.2.1. Web フォームの BADUI

Web 上や現実空間において、ユーザが何かしらの目的を遂行するためにオブジェクトをどのように操作すればよいか分からない、または、操作する際に期待した動作と異なる動作をしてしまうようなユーザインタフェース（以下 UI と表記する）は BADUI (Bad User Interface) [5]と呼ばれている。こうした BADUI は大まかに Web と現実空間のものの2種類に分けられる。その中でも Web ページにおける BADUI の例として図1のように、名前のふりがなをカタカナで入力しなければエラーが出力されてしまう UI が挙げられる。これは、入力すべき文字形式の手がかりとなる文字列（ふりがなの括弧の中や入力例）がひらがなで表記されているため、誤りを誘発してしまう BADUI である。本研究は Web ページの BADUI の中でも例に挙げたような Web フォームの BADUI に着目する。Web フォームの BADUI はユーザの失敗やストレスを招くだけでなく、サービス運営者の利益や信頼を損なう致命的なトラブルに繋がるため問題である[4,6]。実際、過去に存在した療養費支給額を

入力するためのシステムでは、金額の入力ボックスに必ず 13 桁の数値を入力しなければならず、入力したい金額を右詰めにした上で、残りの桁は全て 0 で埋める制約があった（例えば 1351 円支給したい場合、「0000000001351」と入力しなければいけない）。その結果、職員が支給額 1351 円を入力しようとした際、「1351」しか打たなかったため、内部プログラムにより 1 桁目の 1 が消去され、さらに後ろの 10 桁に 0 が加えられたため、支給額が 3,510,000,000,000 円と記載された書面が送付されてしまい、メディアに取り上げられた事例[5]が存在する。



The image shows a web form with two input fields. The first field is labeled '名前:' and contains the text '田中太郎'. The second field is labeled 'ふりがな(かたかな):' and contains the text 'たなかたろう'. To the right of the second field, there is a red error message that reads '入力に誤りがあります！'. Below the input fields is a black button with the white text '送信する'.

図 1 Web フォームの紛らわしい BADUI

1.2.2. BADUI の対策とその問題点

こうした BADUI をなくすため、そもそもの開発段階で BADUI となってしまうことを防ぐ目的とした研究がこれまでに盛んに行われている。具体的には、複数の専門家がサイトのユーザビリティを評価し、問題点を発見する手法や、ユーザビリティを向上させるための汎用的なガイドラインが多く提案されている[7,8]。さらに、Web フォームに焦点を当てたガイドラインも提案されており、その中では Web フォームにおけるユーザビリティの重要性が記述されている[11-13]。実際、Web フォームはユーザビリティの改善を行うことで、ユーザが Web フォームの一連の操作を完了する割合が 10~40%増加する可能性が明らかになっている[2]。しかし、こうした研究が有用であるにも関わらず、サービス運営者の予算やリソースの不足、ユーザビリティを重視しない運営者の存在、経営者のメンテナンスへの意識の低さなどの理由により、BADUI は改善されずに放置されることも珍しくない。

先述した通り、BADUI はサービスの運営者に放置されてしまうことが多い。そこで、エンドユーザの手で BADUI の改善を可能とする手法や、システムを提案する研究が行われている。具体的には、あるユーザが他のユーザのために、操作の仕方を記載した付箋の形をした要素（以下、付箋型アノテーションとする）を用いて、BADUI における注意事項を提示可能とするシステムが提案されている[31,32]。しかし、付箋型アノテーションは Web フォ

ームにおけるユーザの誤入力を誘発してしまう BADUI の改善に特化した機能ではない。また、Web フォームに注意事項を載せた付箋を貼ることが実際に有用であるかどうかは明らかになっていない。

そこで、我々はこれまでの研究[35]において、Web フォームの BADUI を対象としてユーザの失敗やストレスを減らすため、入力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 を実装してきた。このシステムは Web フォームにおける入力内容の自動修正（変換フィルタ）、入力例の提示（入力例付与）、付箋型アノテーションによる操作上の注意事項の提示（ウェプラ）といった複数の改善機能を組み合わせることにより、単一の機能では改善不可能な BADUI におけるユーザの操作支援を目的としている。しかし、WePatch の問題点として以下の 2 つ挙げられる。

1. 改善機能の有用性が明らかになっていない：具体的には、変換フィルタや従来の付箋型アノテーションといった機能は、ユーザのエラー数等の定量的な負担を減らすことが可能かどうか未検証であった。
2. Web フォームの BADUI の改善操作がユーザにとって手間：具体的には、ユーザが Web フォームの入力中に BADUI を発見した場合、ユーザは入力操作を中断する、または入力完了のページ遷移後に、再度フォームのページに戻るといった手順を取る必要がある。

1.3. 本稿の目的

前節の問題点を踏まえ、本稿の第 1 の目的は、変換フィルタや従来の付箋型アノテーション機能が Web フォームの BADUI を改善可能であるかどうかを、ユーザのエラー数やタスク完了時間といった定量的な値を用いて明らかにすることである。第 2 の目的は、ユーザの Web フォームにおける試行錯誤の操作から WePatch の改善機能を推薦可能とする仕組みを実現することで、WePatch の改善操作の手間を減らすことである。

以下 2 章では、本稿の対象である Web フォームの BADUI の原因や分類について詳しく記述する。3 章では開発者による BADUI 検知やユーザによる BADUI 改善、ユーザビリティガイドラインに関する研究について述べ、本稿の位置付けを明確化する。4 章では BADUI の Web フォームでの入力タスク実験により、WePatch の改善機能の有用性を定量的に評価する。次に、5 章において改善操作の手間を減らすため、半自動的な BADUI 改善手法を提案し、変換フィルタにおいて半自動 BADUI 改善手法を実現したプロトタイプシステム「WePatch 2」について述べる。6 章では WePatch 2 をユーザベースの実験により、Web フォームのユーザビリティが向上するかどうかを検証する。7 章では 6 章から得られた WePatch 2 の課題について改良を行なったシステム「WePatch 2+」がどの程度 BADUI を検知可能であるかについて検証実験を行い、システムの有用性について記述する。最後に 8 章でまとめと WePatch 2+ のリミテーション、今後の応用先を記述する。

第2章 Web フォームの BADUI とその分析

ここでは、Web フォームにおいてユーザが誤った文字形式で入力してしまう BADUI について記述する。まず、Web フォームはユーザがデータを送信するための UI、データベースに値を受け渡すためのプログラム、入力されたデータを格納するためのデータベースの3つのレイヤーに分けられる。この中でも、内部のプログラムとデータベースが求める文字形式に関する情報は UI から見えない部分である。そのため、内部プログラムとデータベースが求める文字形式と、UI が求めている入力形式が矛盾している場合や欠如している場合に BADUI となることが多い。具体的には、図2の UI 側の住所の入力項目では、入力例の文字列から半角入力も許容すると予測できる。しかし、その通りに入力すると、内部では全角入力のみ通すようなプログラムとなっているため、送信エラーが出力されてしまう例が挙げられる。これは、ユーザから半角入力が可能に見える UI と、ユーザからは見えない内部の全角入力を要求するプログラムやデータベースとの間に矛盾が生じているためと言える。

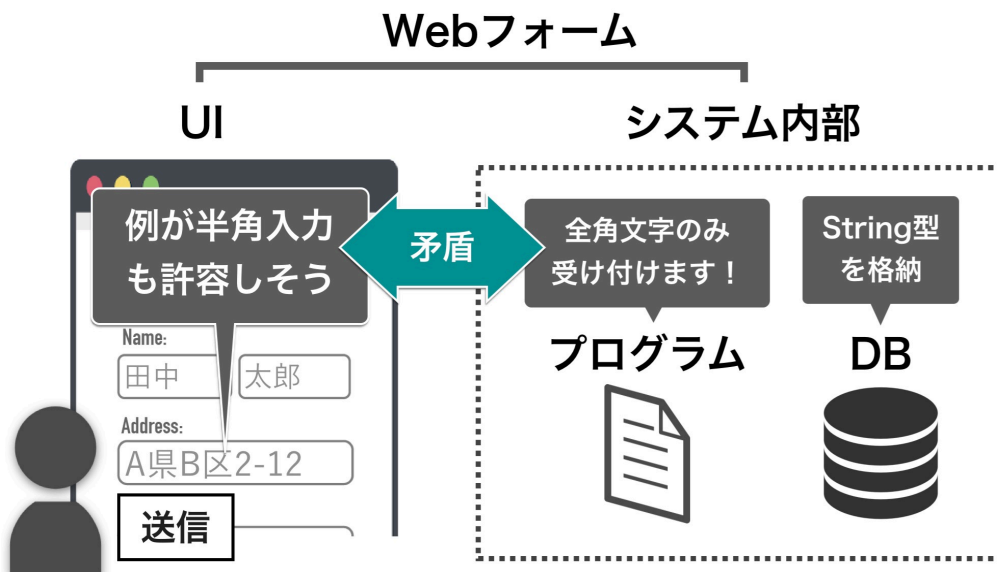


図2 Web フォームが BADUI となる原因

そもそも Web フォームが BADUI となる原因の例としては、図3のような名前のフリガナの姓名の順番が慣習とは逆になっているような BADUI に着目すると、開発者が過去に自社で利用していたデータベースや、海外の会社で利用されていたデータベースを再利用して新たな Web フォームを制作する場合である。開発者が UI やプログラム側でデータベースに合うようなシステム構築を行った結果、利用対象のユーザを混乱させてしまうものとなったと考えられる。また、単に開発者が UI 側の入力例においてカタカナで表記するところを誤ってひらがなで表記するといった誤字が原因で、UI と内部プログラムの間にギャップが生じてしまう場合も考えられる。

氏名 (本人確認証 (パスポート、日本の運転免許証、日本の在留カード又は日本のマイナンバー))

* 姓 (日本国籍の方は必須)

* 名 (日本国籍の方は必須)

英字氏名 (パスポートをお持ちの方は、表示名のアルファベット半角・大文字で記入)

* 名 (アルファベット半角・大文字で記入)

* 姓 (アルファベット半角・大文字で記入)

ミドルネーム (アルファベット半角・大文字で記入)

図3 姓名の入力順番に一貫性がないBADUI

次に、Web 上の膨大な数の誤入力を引き起こすBADUIをどういった手段で改善するか、また改善可能性を明確化するため、誤入力を引き起こすBADUIと入力ボックスの制約について分類を行う。分類にあたって、まず、これまで収集された膨大なBADUIからユーザの誤入力の4種類の原因について記述する。

- 誤った入力形式に誘導してしまうUI (ミスリード)
- 入力形式に関する情報が欠落しているUI (手がかり欠如)
- ユーザにとって馴染みのない入力形式を求めるUI (慣習との乖離)
- 入力操作において過度な労力を強いるUI (一苦労)

また、BADUIには図1のBADUIのように、名前のふりがなを誤ってひらがなを入力してしまうような、単一の入力ボックスを構成する要素(入力項目名、入力例等)が原因で引き起こされるもの、また、図4に示すような郵便番号の項目ではハイフンが不要と表記されているにもかかわらず、電話番号などの項目ではハイフンが必須であるような一貫性が欠如しているような、前後の入力ボックスの構成要素の関係が原因で引き起こされるものの2種類が存在すると考えた。以上の失敗原因の4分類、単一と複数間のBADUIの違いによる2分類から、Webフォームにおいて誤入力を引き起こすBADUIは表1のように分類できる。なお、表1の先頭列のユーザが誤った文字形式を入力してしまう原因の種類を表しており、単一は特定の単一の入力ボックスが原因で誤った文字形式を入力してしまうもの、複数間は複数の入力ボックスがあるとき、その関係性により誤った文字形式を入力してしまうものを表している。

メールアドレス※	<input type="text"/> (半角)
メールアドレス再入力※	<input type="text"/> (半角)
ご住所※	〒 <input type="text"/> 例) 3960066 - (ハイフン) 不要 都道府県 <input type="text"/> <input type="text"/> ※市区町村、番地、建物名、部屋番号まで
電話番号※	<input type="text"/> 例) 078-0000-0000
FAX番号	<input type="text"/> 例) 078-0000-0000
携帯電話	<input type="text"/> 例) 090-0000-0000

図4 ハイフン含むかどうかが一貫していない BADUI

表1 入力形式に関する BADUI 分類

	単一の入力ボックス	複数の入力ボックス間
ミスリード	内部が求める文字形式と異なる表記 (例：図1と同様)	前後の入力形式に一貫性がない (例：図4と同様)
手がかりの欠如	内部が求める文字形式を表記していない (例：郵便番号にハイフンを含めなければいけないが、注意書き等が表記されていないもの)	入力の手がかりが不十分 (例：住所の入力欄が「住所1」と「住所2」と分かれているが、それぞれの項目に書く内容が表記されていないもの)
慣習との乖離	大多数のユーザにとって慣れない入力形式を要求するもの (例：誕生日をスラッシュではなくハイフン区切りを要求するもの)	入力順番が慣習と異なるために紛らわしいもの (例：図3と同様)
一苦労	特殊な入力形式を要求するもの (例：名前のフリガナにおいて半角カタカナでの入力を要求するもの)	余計な入力項目を設けているもの (例：メールアドレスとパスワードに加え、必要ないにも関わらず ID も要求するもの)

また、入力ボックスの制約については、下記の5つに分類できる。

- **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，過去に実装した WePatch[35]と5章において記述する WePatch 2 の改善可能な入力形式に関する BADUI 分類は「ミスリード（単一）」「ミスリード（複数）」「手がかりの欠如（単一）」「慣習との乖離（単一）」「一苦勞（単一）」であり，改善可能な入力ボックスの制約は「文字種の制約」，「記号に関する制約」，「半角・全角」である。

第3章 関連研究

3.1. UI デザインガイドライン

Web 上でユーザが BADUI によって失敗や悩みを引き起こさないよう、そもそも Web ページの制作の時点でユーザビリティを考慮した UI 設計を行うことが重要である。これまで過去の膨大な Web サイトの UI における失敗事例の知見が集約されたことで、開発者が守るべきガイドラインは数多く提案されている。Nielsen ら[7]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。Bailey[8]は既存のユーザビリティに関するガイドラインが、一般的な Web デザイナーが利用するためには難しいものであるため、Web サイトの品質が不均一になってしまうことを問題とし、理解しやすく実用的な形式で利用可能なガイドラインを提案した。IBM[9]は HTML のタグの利用方法に従うことで Web 標準に従った Web サイトを制作可能なページを公開している。また、Apple[10]は専門家の知識を集約した Web サイトのデザインの提案を行なっている。しかし、こうしたガイドラインの確認項目は分かりやすいが抽象的であることや、Web サイトの種類によってはユーザビリティの確認を網羅しきれない側面があるため、Web サイトの種類ごと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、Avila[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、Seckler[12,13]らによって、ガイドラインが実際に有用であることが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利用せず開発することで、BADUI が作成されてしまうことは少なくない。本稿はこうした BADUI を発見するための手法やガイドラインから漏れてしまった BADUI や、放置されてしまう BADUI の問題を解決することを目的としている。

3.2. 人手によるユーザビリティ評価

これまでに Web サイトのユーザビリティの問題点の特定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 1 つとしてヒューリスティック評価法[14,15]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、より幅広い意味でのユーザビリティの評価尺度を用いた SUS (SystemUsabilityScale) [16]では、低コストで信頼性の高いユーザビリティ評価を可能としている。さらに、本稿の改善対象である Web フォームのユーザビリティ評価を行うための FUS (FormUsabilityScale) [17]というアンケートが提案されている。このアンケートによって Web フォームのユーザビリティを手軽に測定可能である。また、Web ユーザビリテ

ィ評価を効率的に行うためのツール[18,19]が公開されており、開発者がユーザビリティ評価を行うための環境は整っていると言える。しかし、これらのユーザビリティの問題点を早期発見可能な手法が存在するにも関わらず、ユーザビリティを重視しない運営者の存在により、BADUIは放置されてしまうことが多い。また、これらは手軽に行える手法であるが、多くの人手を要することから同様の理由で評価が実施されないことは少なくない。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、放置されてしまっているBADUIであっても改善可能という利点がある。

3.3. 自動UI分析

人手によるユーザビリティ評価にかかるリソース不足の解消や、人手では発見することが難しいユーザビリティの問題点の発見を目的として、自動的なUI分析やBADUI検知を目的とした研究はこれまで盛んに行われている[20]。具体的には、人手による評価での発見が難しい潜在的な問題を、ユーザのページ遷移やページ内移動、インタラクションを可視化することによって発見を促す研究が多く行われている[21-25]。さらに、画像認識技術や検出モデルの制作によってBADUI検知を試みる研究も行われている[26]。また、クリックやスクロール等の操作ログを取得し、ユーザビリティの問題点を自動的に検出する研究もいくつか提案されている。具体的には、ユーザのイベントデータをクライアント側で取得、データ処理を行い、自動的にユーザビリティの問題点の検知を可能とするシステムが開発されている[27-30]。このようなシステムでは、開発者はシステムから送られてきたログデータを参考にBADUIを改善可能である。

これらの研究において、ユーザのページ上の操作を分析することでBADUIを検知している点については我々の手法の目指すところと同じであるが、我々は入力操作からBADUIを検知することに焦点を当てている点において異なる。また、Webフォームの検知から改善まで行えるという点が異なる。

3.4. ユーザの操作支援

Web上において、ユーザが誤った操作を事前に防ぐことや、誤った行動の後に自動的に操作を修正することで、BADUIにおける操作を支援する研究は数多く行われている。その中でもユーザ自らWebサイトを拡張する技術の実現が行われるようになった[31,32]。代表的なものとしては、ユーザがWebサイトに操作時の注意事項が記載された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が実現されている[33,34]。付箋型アノテーションはWePatchのウェブラと類似した機能であるが、我々のシステムは複数の改善機能を組み合わせることで、より多くの種類のBADUIが改善可能になるという点で異なる。また、そもそも付箋型アノテーションがWebフォームにおけるユーザの負担を

減らすかどうかは明らかになっていないため、本稿では定量的にウェブや入力修正機能などの改善機能がユーザの負担を減らすかどうか定量的に検証し、従来の付箋型アノテーションの有用性を明らかにする。

Dong[36]らは、Web サイト運営者のユーザビリティ評価によるリソース不足を解消するため、ユーザのコミュニティが 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行うことによってサイトの文字列を拡張し、ユーザ同士で操作を支援可能とするシステムを実現している。また、その他にも文字列へのハイパーリンクの付与や、メモ代わりのアノテーションの付与といった拡張機能が多くのシステムで実現されている[37-40]。これらの研究では主に Web 上のテキストを読むことを中心に行うページにおいて、誤った操作をしやすい要素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。Sergio[41]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可能であるが、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ためリアルタイムにユーザを支援する必要がないという利点がある。Swearngin[42]らはユーザの聴覚や視覚等に問題がある場合であっても、Web サイトの操作方法を音声入力やジェスチャー入力といった別のインタラクションに切り替えることで、アクセシビリティを改善可能とする拡張機能を開発している。Swearngin らは身体的なアクセシビリティの問題を解決することを目的としているが、我々の研究は Web サイトの使いにくさ、分かりにくさを改善することを目的としている点で異なる。

Web フォームはユーザにとって入力量が負担になるものが多いという問題点から、ユーザの入力ボックスに全ての単語を打ち込む前に、打ち込む単語を予測し、入力を支援する研究が行われている[43,44]。これらの研究はユーザの入力量を減らす目的であるが、我々の研究は誤った操作を引き起こすような Web フォームを改善することで、ユーザの失敗を減らすことが目的であるという点で異なる。

第4章 WePatch の評価実験

4.1. WePatch

我々はこれまでに Web 上の改善されていない BADUI をユーザの手で改善可能とするシステムを WePatch と名付け、JavaScript と PHP を用いて Chrome 拡張として開発してきた。また、これまでの評価実験での結果や実験協力者の意見から WePatch が導入されたときの UI や機能の改良を行なってきた。本章では Web フォームにおける WePatch の有用性を検証するため、問題の多い入力フォームの改善に有効であると考えられる変換フィルタ、入力例付与、ウェブラの3つの機能を評価対象とする。機能の詳細については[35]に記述しているが、以下にこの3機能についてのみ簡潔に説明する。

- **変換フィルタ**: 入力フォームへの入力内容を、自動的に適切なものに変換することによりミスを防止する。機能には、半角から全角（全角から半角）への変換、ひらがなからカタカナ（カタカナからひらがな）への変換、指定した文字列が含まれているときに自動的な除去を行うフィルタがある。図5は付与された変換フィルタによって入力された文字列が正しく変換される様子である。

The figure consists of two vertically stacked screenshots of a web form. The top screenshot shows a form with two input fields: '住所' (Address) and '電話番号' (Phone Number). The '電話番号' field contains the text '080-1234-5678'. Below the fields are two buttons: '送信' (Send) and '1つ戻る' (Back one). The bottom screenshot shows the same form after a transformation. The '電話番号' field now contains '08012345678'. A tooltip with the text '修正しました' (Corrected) points to the phone number field. The '住所' field is empty. The '送信' and '1つ戻る' buttons are still present.

図5 変換フィルタ

- **入力例付与**: 入力フォームに入力例を付与し、入力すべき情報をわかりやすくする。図6はユーザがその入力フォームに適した入力例を付与し、それを共有することで他ユーザの画面に入力例が提示されている例である。

Date of Birth <input type="text"/>	Date of Birth <input type="text" value="1999/01/01"/>
Email address <input type="text"/>	Email address <input type="text" value="s.sample.000@sample.cc"/>
Home Phone <input type="text"/>	Home Phone <input type="text" value="023456789"/>
<input type="button" value="Sign up now"/>	<input type="button" value="Sign up now"/>

図 6 入力例付与

- ウェブラ：付箋型のアノテーションを貼り付けることで手掛かり情報を提示する。図7は入力フォームの確認ボタンの下に承諾済みのメールマガジンの受信設定の要素が隠れており、そのことをユーザに注意している例である。

※メルマガの購読を希望されない方はページ下部の「メルマガ配信」をご確認下さい

↓メルマガ設定あります！

予約内容を確認する

メールマガジン受信設定

入力された会員ID（メールアドレス）に会員様様定のお知らせメールをお送りいたします。
配信を希望されない場合は、チェックをお外しください。

三越伊勢丹×リクルートのギフト通販サイト「キノギフト」 キノギフト 利用規約

キャンペーン・ポイントなどのお得情報や、季節のギフト情報をお届けします。
メール配信の変更・削除は「キノギフト」よりお願いします。
PC、スマートフォンでのみ対応となります

図 7 ウェブラ

4.2. 実験準備

ここでは、WePatch の改善機能によってユーザの BADUI における負担がどの程度軽減されるかを実験により明らかにする。具体的には、システムによる改善前と改善後のそれぞれの Web フォームにおいて実験協力者に入力タスクを課し、その過程で得られるデータを比較する。実験のため、我々は特設の Web フォームを 4 種類（図 8~11）作成した。なお、表 2 に作成した実験用の Web フォーム A~D とそれぞれに含めた BADUI を記述する。

お名前 (ふりがな)	<input type="text"/>
	例) たぐたろう
メールアドレス(PC)	<input type="text"/>
	例) sample@smail.com
ご住所	郵便番号 <input type="text"/>
	(ハイフンあり)
	都道府県の選択 <input type="button" value="v"/>

図 8 実験用 Web フォーム A

フリガナ※	<input type="text"/>	（姓と名の間に全角スペースをいれてください、ひらがな入力）
性別※	<input type="radio"/> 女性 <input type="radio"/> 男性	
生年月日※	<input type="text"/>	(yyyy/mm/dd形式)
ご住所※	〒 <input type="text"/>	(ハイフンは必要ない)

図 9 実験用 Web フォーム B

お名前など	
お名前 (漢字)	せい 漢字 <input type="text"/>
	めい 漢字 <input type="text"/>
よみがな(かたかな)	せい よみがな <input type="text"/>
	めい よみがな <input type="text"/>
電話番号(自宅)	<input type="text"/>
	<input type="button" value="080 1234 5678"/>

図 10 実験用 Web フォーム C

<ul style="list-style-type: none"> ●さんぶるてきすとメニュー ●さんぶるてきすとメニュー ●さんぶるてきすとメニュー 	生年月日(yyyy/mm/dd) <input type="text"/>
	メールアドレス(携帯) <input type="text"/>
	郵便番号 <input type="text"/>
	住所 <input type="text"/>

図 11 実験用 Web フォーム D

各 Web フォームには変換フィルタ、入力例付与、ウェブラの 3 機能の改善対象であるミスリード（単一）、ミスリード（複数）、慣習との乖離（単一）、手がかりの欠如（単一）の BADUI を用意した。なお、これらは BADUI タレこみサイト[45]や実際の Web サイトに存在していたものを参考にしたものである。ここで、4 種類の実験用 Web フォームに対して、先述の変換フィルタ、入力例付与、ウェブラの 3 機能をそれぞれ単体で用いて改善を行い、改善前のものを合わせて合計 16 種類の実験用 Web フォームを用意した。これらの改善機能の付与は用意した BADUI に対して適切であると考えられる箇所に対して著者自身が行った。各機能による改善例を図 12~14 に示す。図 12 は住所において半角から全角へ変換を行うフィルタ、電話番号ではハイフンを除去するフィルタが付与されている様子である。図 13 は名前の姓と名の上に空白が必要なことがわかるような入力例を付与している様子である。図 14 は Web フォーム内の注意書きや入力形式に関する手がかりに注目するように促している様子である。

4.3. 実験手順

次に、実験協力者 16 名（普段からパソコンを使用する 20~23 歳の男性 13 名、女性 3 名）を集め、実験用 Web フォームでタスクを行ってもらった。ここで、各実験協力者を 4 人 1 グループ 1~4 に分け、各グループに対して割り当てられた改善後（または改善前）の Web ページを表 3 に示す。Web フォーム A~D は作成した 4 種類の実験ページを表している。また、Web フォーム A~D でタスクを行ってもらう際の順番は無作為に決められている。

実験協力者には実験概要として Web の登録フォームで入力タスクを行ってもらうことを説明した上で、こちらが事前に用意したダミープロフィールの人物として、それを参考にしながら入力するよう依頼した。また、WePatch の改善機能によって、どのように入力の修正や手がかりの提示が行われるかについて説明した。さらに、入力中の注意事項として、質問項目の全てに回答すること、入力ボックス間の移動はクリックのみで行うこと（タブキーでの移動ができないよう制御した）を伝えた。なお、実験協力者には BADUI のようなユーザの失敗を誘発するものが配置されていることを説明せず、普段 Web フォームを利用するときと同じように入力を行ってもらった。この理由としては、実験協力者が入力の手がかり等を普段以上に注意しながら入力してしまうことを可能な限り防ぐためである。

入力タスク中には実験協力者の Web フォームにおける負担を測るため、各実験用 Web フォームにおける「テキストボックスへのフォーカス（テキストボックスがクリックして選択）された回数」「タスクを行っていた時間（秒）」「タイピング総数」「入力確定ボタンを押した際、情報に不備がある場合に出力されるエラーフィードバック回数」の 4 種類のデータを取得した。これらのデータ取得を行う Chrome 拡張を実験協力者の PC 環境にインストールしてもらうことで入力タスク中に自動的にデータを取得した。

表2 Web フォームに含めた BADUI 一覧

フォーム	BADUI	詳細
A	ミスリード (複数)	メールアドレスや郵便番号は半角入力であるのに関わらず、住所は全角入力をしなければならない
	ミスリード (複数)	郵便番号ではハイフンが必要であるのに関わらず、電話番号ではハイフンを入れてはいけない
B	慣習との乖離 (単一)	名前の姓と名の上に全角のハイフンを入れなければならない
	ミスリード (複数)	電話番号ではハイフンを入れなければならないのに関わらず郵便番号では入れてはいけない
C	慣習との乖離	名前の姓と名の上に全角のハイフンを入れなければならない
	ミスリード (複数)	電話番号ではハイフンを入れなければならないのに関わらず郵便番号では入れてはいけない
	ミスリード (単一)	名前のふりがなをカタカナで入力する必要があるにも関わらず、ラベルや入力欄の中がひらがなで「よみがな」と書いてある
	ミスリード (単一)	入力例には郵便番号の間やカード番号の間に空白が書いてあるにも関わらず、空白は不要である
	ミスリード (複数)	メールアドレスや郵便番号は半角入力であるのに関わらず、住所は全角入力をしなければならない
D	手がかりの欠如 (単一)	入力例が書かれていないが入力形式をユーザは推測して入力しなければならない
	手がかりの欠如 (単一)	ラベルにフリガナという表記のみあり、カタカナ入力しなければならない

表3 実験ページの割り当て

	入力 フォーム A	入力 フォーム B	入力 フォーム C	入力 フォーム D
グループ1	改善なし	入力例付与	ウェブラ	変換フィルタ
グループ2	ウェブラ	改善なし	変換フィルタ	入力例付与
グループ3	入力例付与	変換フィルタ	改善なし	ウェブラ
グループ4	変換フィルタ	ウェブラ	入力例付与	改善なし

ご住所	都道府県の選択 ▼
	<input type="text"/> 例) 千代田区大手町1-2-3〇〇ビル10F
電話番号(自宅)	<input type="text"/> (ハイフンなし)
電話番号(携帯)	<input type="text"/> (ハイフンなし)
FAX番号	<input type="text"/> (ハイフンなし)
会社名/団体名	<input type="text"/> 例) 株式会社 タグインデックス

図 12 変換フィルタによる改善例

お名前※	<input type="text" value="田中 太郎"/> スを入れてください)
フリガナ※	<input type="text" value="たなか たろう"/> スを入れてください、ひら
性別※	<input type="radio"/> 女性 <input type="radio"/> 男性

図 13 入力例付与による改善例

※郵便番号や電話番号の空白文字は不要です ←この注意事項を必ずチェックしてください

お名前など

お名前 (漢字)

よみがな(かたかな)
 よみがなは「カタカナ」入力です

電話番号(自宅)
 例) 080 1234 5678

図 14 ウェプラによる改善例

4.4. 結果

まず改善前の各 Web フォームにおいて入力タスクの過程で得られたデータを表 4 に示す。表 4 のフォーカス数は「テキストボックスへのフォーカス（テキストボックスがクリックして選択）された回数」、時間（秒）は「タスクを行っていた時間（秒）」、タイプ数は「タイピング総数」、エラー数は「入力確定ボタンを押した際、情報の欠如や入力形式の違いがある場合に出力されるエラーフィードバック回数」をそれぞれ表している。

次に、改善前と改善後の比較データを図 15~18 に示す。それぞれのグラフの横軸は取得したデータを表しており、縦軸は改善前のデータを 1.00 としたときの比率を表している。また、灰色の棒グラフは改善前、青色は変換フィルタ、オレンジ色は入力例付与、黄色はウェブラを用いたときの取得したデータの改善前に対する比率を表している。

Web フォーム A ではエラー数を減らした入力例付与と、エラー数以外の負担を軽減した変換フィルタが有効であった。それに対し、ウェブラはエラー数を増やしてしまい、負担を増加させる結果となっていた。Web フォーム B ではウェブラ機能がフォーカス数やタイピング数といった負担を軽減しているものの有効な機能は特になく、改善機能を用いたときにタスク時間やエラー数が増加しているため、負担を増加させていた。Web フォーム C では変換フィルタによってエラー数やタスク時間が大きく減少していることがわかる。一方、ウェブラについてはエラー数を減少させているもののタスク時間が増加してしまっていた。Web フォーム D では変換フィルタを用いたときにタスク時間がやや増加したもののエラー数やフォーカス数が減少しており、全体としては負担の軽減に有効であると考えられる。一方で、入力例付与とウェブラは全ての値が増加しており、特にウェブラが負担を増加させてしまっていることがわかる。さらに、図 15~18 の結果をデータごとに平均化したものを図 19 に示す。グラフより、変換フィルタが Web フォームの BADUI の改善に有効であることが明らかになった。一方、ウェブラや入力例付与は逆にユーザの負担を増加させてしまうことが明らかとなった。

表 4 改善前のデータの平均値

	入力 フォーム A	入力 フォーム B	入力 フォーム C	入力 フォーム D
フォーカス数	13.75	24.25	25.50	10.25
時間（秒）	223	314	442	171
タイプ数	282	383	386	220
エラー数	0.25	1.25	2.25	1.00

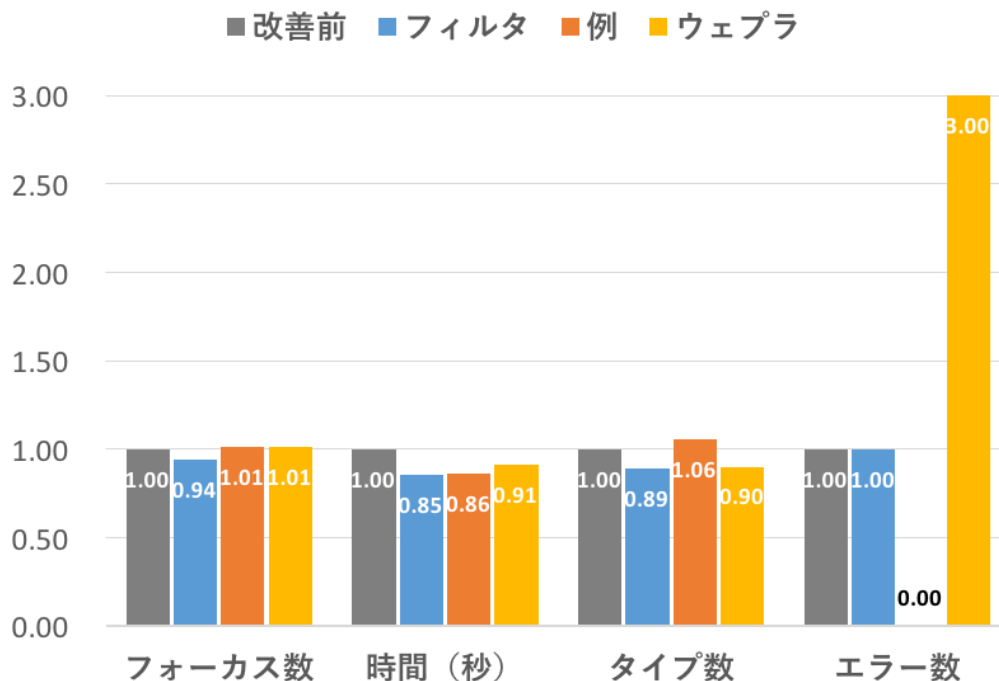


図 15 Web フォーム A における結果

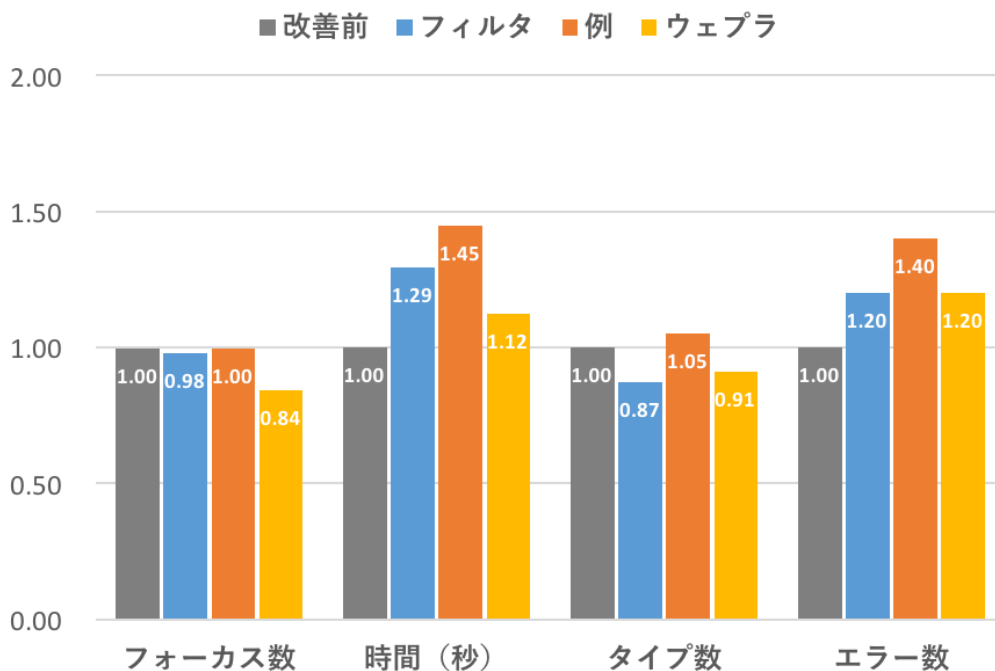


図 16 Web フォーム B における結果

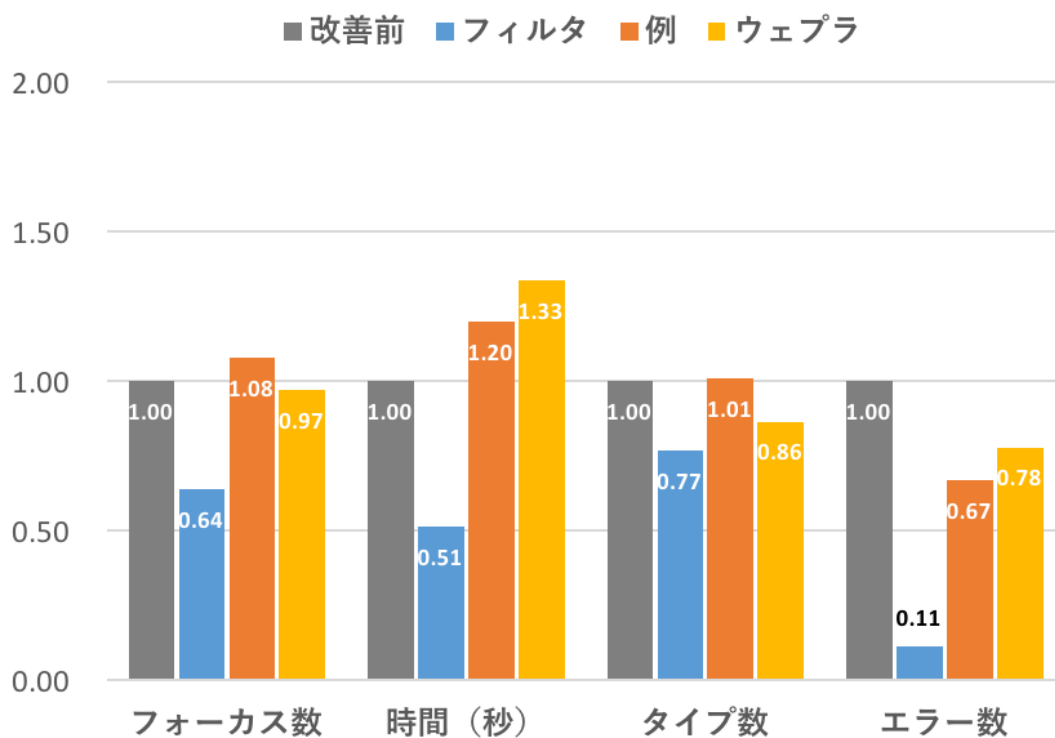


図 17 Web フォーム C における結果

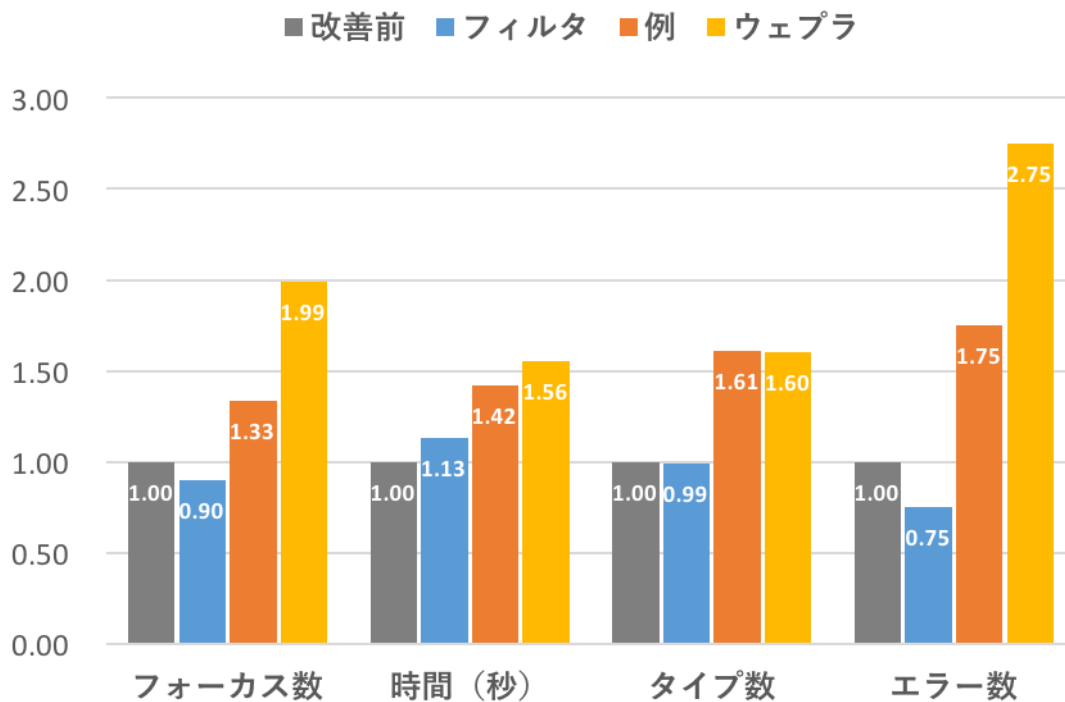


図 18 Web フォーム D における結果

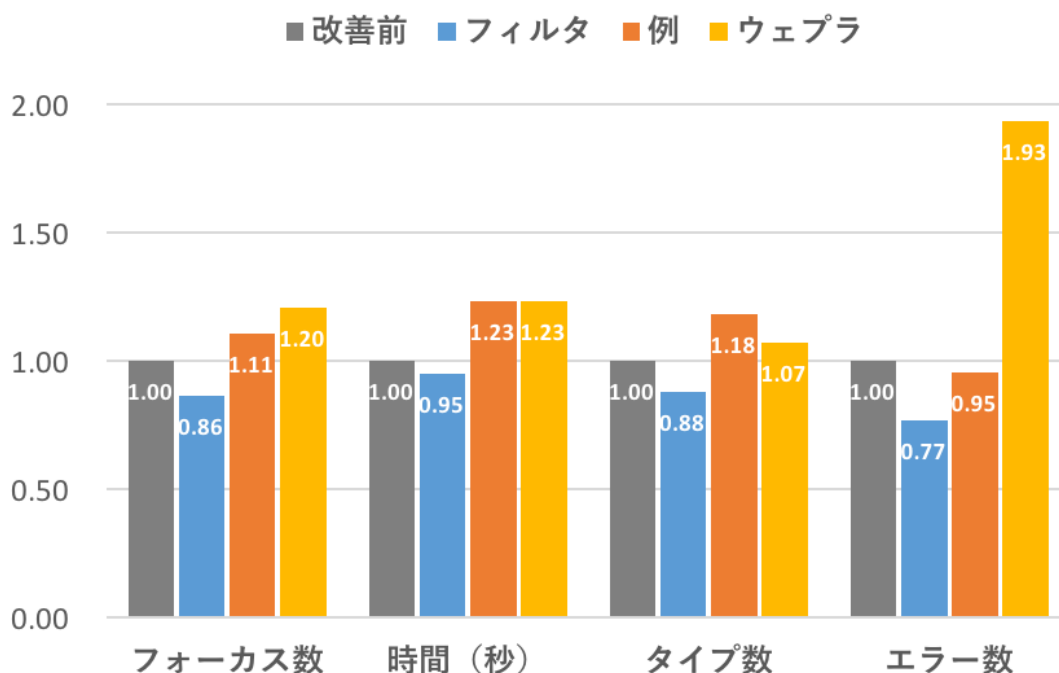


図 19 Web フォーム A~D の平均結果

4.5. 考察

実験結果より、変換フィルタによって多くの Web フォームの BADUI におけるユーザの負担を減らすことが可能であると考えられる。また、変換フィルタはユーザの負担を減らすことができなかった場合であっても、入力例付与やウェプラ機能のように大きく負担を増やしてしまうといった可能性は低いと言える。一方、従来の研究[33,34]ではウェプラ機能のようなアノテーションはユーザの Web ページにおける操作の負担を軽減可能であると考えられてきたが、本章の評価実験により、付箋型アノテーションが万能な機能でないことが明らかになった。付箋型アノテーションが負担を大きく増やしてしまう原因としては、ユーザが入力中にアノテーションを気にかけることで、無意識に負担をかけるためであると考えられる。こうした、付箋型アノテーションのような追加的なものが効果的に働くかどうかについては、現実世界において貼られているテプラが本当に有効なものなのかどうかを議論するうえでも重要になる。そこで、今後はさらに踏み込んだ実験を実施することにより、その有用性について検討する必要があると考えられる。なお、本章の評価実験ではユーザの誤った形式での入力を誘発するような BADUI のみを用いていた。そのため、例えば図 20 のように、慣習的に入力確定ボタンを配置する位置に入力内容のリセットボタンが配置されており、ユーザが入力内容のリセットする意図が無くても誤ってリセットボタンを押してしまうような BADUI に対しては有効であると考えられる。実験では有効でなかった入力例

付与については、今回実装した方法は、ユーザが何か入力するまでは灰色の文字で例を表示しているものの、ユーザが何らかの入力を行うと消えてしまうというものであった。そのため、実験協力者は例を確認せずに入力を開始してしまい、元の例がわからなくなってしまった結果、効果が薄くなってしまったと考えられる。これらのことから、ユーザの手による Web フォームの BADUI を改善するためには、ユーザが入力する前に文字を使って注意を喚起するのではなく、誤入力後に自動的な修正をすることが負担の軽減に最も有効であると考えられる。

ここで、WePatch の課題として、ユーザがシステムを用いて BADUI の改善を行うためには、Web フォームにおいてユーザは登録を中断するか、または登録が完了してから再度フォームのページに戻るといった手順を取る手間が挙げられる。これは、ユーザの積極的な BADUI 改善への参加を妨げてしまう要因となるため、ユーザがより手間なく改善可能にする必要があると考えられる。ここで、2章で述べたような UI と内部プログラムやデータベースの求める入力形式が矛盾している点に着目する。我々はこうした BADUI において、ユーザの不正解の入力形式から正解の入力形式に修正するといった行動を利用することで、BADUI を検知し、その入力フォームが求める入力形式に自動修正する変換フィルタを推薦可能になると考えた。そこで、2章と本章の知見から、5章ではユーザの手による BADUI 改善をより手間のないものにするため、ユーザの入力の試行錯誤を利用した半自動的な BADUI 改善手法を提案する。

The image shows a registration form with the following fields and labels:

- 電話番号 ※ (半角) 【例 0120123XXXX】
- 性別 男性 女性
- 生年月日 ※ [] 年 [] 月 [] 日 (年は西暦) 【例 1951年 1月 1日】
- メールアドレス ※ ← PC/スマートフォンで受け取れるメールアドレスでのご登録をお願いします。
- メールアドレス再入力 ※ ← コピーせずに確認のためもう一度入力してください。
- パスワード ※ (英数字を混ぜた半角英数字) 【4文字以上16文字以内】
- パスワード再入力 ※ ← コピーせずに確認のためもう一度入力してください。

Below the form fields, there is a button labeled "入力内容をリセット" (Reset input content). A yellow callout box points to this button with the text "←これはリセットボタンです！" (← This is the reset button!).

Below the form, there is a section titled "■シネマシティWeb予約利用規約" (Cinema City Web Reservation Terms of Use). It contains the following text:

シネマシティWeb予約(以下、当サービス)をご利用いただく場合には、シネマシティWeb予約利用規約(以下、当規約)が適用されます。当サービスを利用することによって、当規約の内容を承諾したものとみなします。全条件事項に同意できないユーザーは、当サービスの利用をお断りいたします。ご注意ください。

【サービス内容】

シネマシティWeb予約利用規約に同意します。

上記規約に同意される場合は、チェックをつけて会員登録ボタンをクリックしてください

図 20 リセットボタンの BADUI とウェブラによる改善

第5章 WePatch 2

5.1. 提案手法

既存のユーザの手による Web 上の BADUI 改善手法を手間が少ないものにするため、ユーザの失敗および正解行動からその Web フォームが本来求めている文字形式を推定し、適切な改善機能の推薦を行う、半自動 BADUI 改善手法を提案する。例えば、ある Web 上の電話番号の入力ボックスでは「数値の間にハイフンを挟む文字列」しか受け付けないが、ユーザがそれに気づかず「数値のみの文字列」を入力して送信ボタンを押したときには、システムは何らかのエラーメッセージを表示する。この時、「ハイフンを含めてください」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、数値にハイフンを含め、送信ボタンを押すと考えられる。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入力形式を修正する。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数値にハイフンを挟む文字列なのではないか?」と推定することで、正しい文字形式（ここでは数値の間にハイフンを挟む文字列）へとたどり着くことができるかもしれない。また、正しくハイフンを追加して入力した後、送信ボタンを押すと、エラーメッセージは表示されず、サービスへの登録等が完了する（図 21）。

本提案手法の目的はこうした 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、失敗と成功の差異から、対象の入力ボックスに対して適切な変換フィルタ（この場合は、数値のみ→数値の間にハイフンを補完するフィルタ）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 22 に示す。

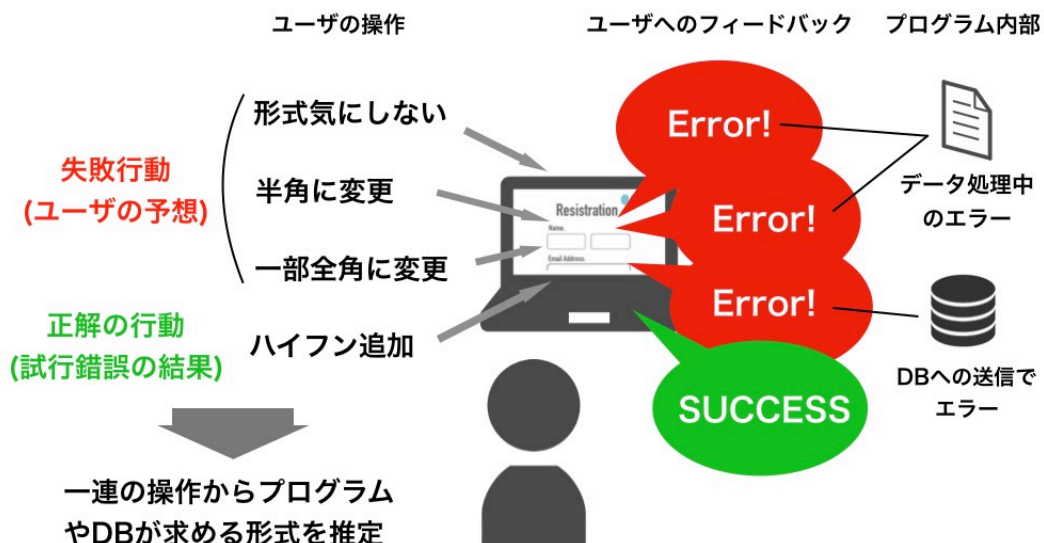


図 21 ユーザのトライアンドエラー

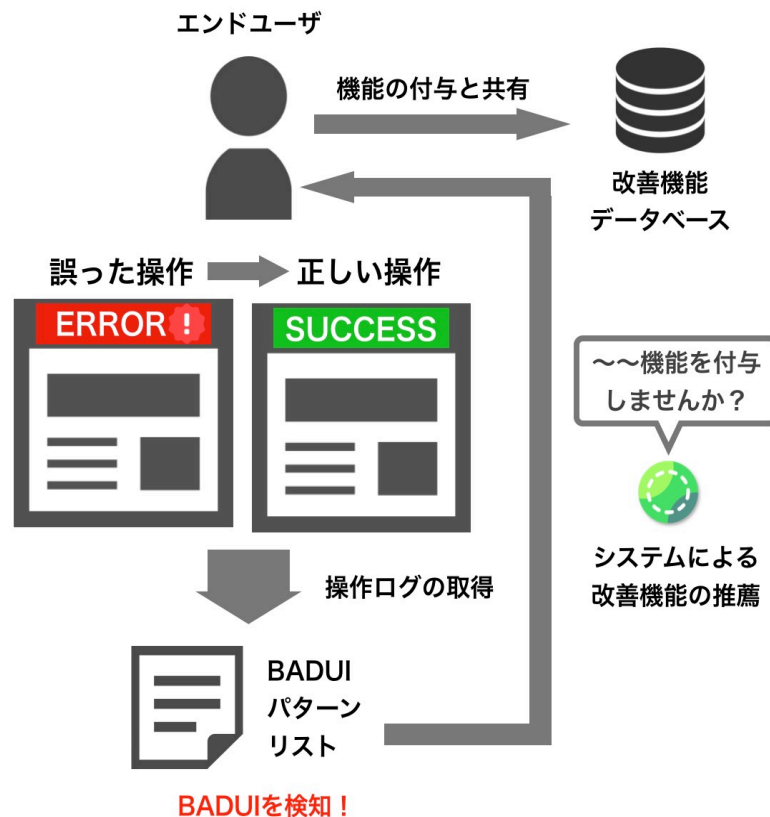


図 22 BADUI 検知と改善プロセス

5.2. Web フォームにおける入力形式の制約

提案手法を実現するため、2章で述べたような文字形式の BADUI の中でもミスリード（単一と複数）、手がかりの欠如（単一のみ）、慣習との乖離（単一のみ）、一苦労（単一のみ）を改善対象とし、また、テキスト入力ボックスの制約の中でも文字種の制約、記号、半角・全角、数値データに関する BADUI を改善可能とする WePatch 2 を実装した。ここでは、WePatch 2 の 3 つの機能と変換フィルタの種類（表 5）を記述する。

- **変換フィルタの推薦**：ユーザが誤った入力（図 23）を行った後、正しい入力操作をすることで BADUI を検知し、ボタンと吹き出しが表示される（図 24）。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。
- **入力の自動修正**：変換フィルタが適用された Web フォームでは、ユーザが入力した文字に対し修正が行われる。修正時は文字が点滅するフィードバックがあり、ユーザは文字の修正が行われていることが把握できる。図 25 は誕生日にスラッシュの記号が入力

後に自動補完された様子を表している。

- **変換フィルタへの評価**：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が把握できる。図 25 は自動補完後に変換フィルタの評価ボタンが提示されている様子である。

図 23 誤った形式での入力

図 24 フィルタ推薦の様子

図 25 変換フィルタによる修正と評価ボタンの提示

表5 WePatch 2の変換フィルター一覧

制約	変換フィルター
文字種の制約	ひらがなからカタカナへ変換 カタカナからひらがなへ変換 英字を小文字から大文字へ変換 英字を大文字から小文字へ変換
記号に関する制約	スペース削除 スペース補完 (名前) ハイフン削除 ハイフン補完 (郵便番号, 電話番号) スラッシュ削除 スラッシュ補完 (誕生日) ハイフンからスラッシュへ変換 スラッシュからハイフンへ変換
半角・全角	半角文字から全角文字へ変換 全角文字から半角文字へ変換
数値データに関する制約	11桁の数字の先頭の0を削除した上で81(国番号)を補完(電話番号)

5.3. 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ー間で反映される。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ー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、入力形式はユーザー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。

第6章 ユーザベース実験

6.1. 実験準備

WePatch2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、WePatch2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。

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、その中か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6に示すBADUIを選定した。

ここで、4章で作成した実験用Webフォームはあまりに入力項目が多く、1ページあたりの負荷が結果に大きく影響してしまうと考えたため、本章で用意する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。Webフォームの画面では、入力項目名、文字形式の手がかり、入力例を提示した。具体的な入力項目は、名前に関する情報(漢字、ひらがな)、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。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。図26は、実験用のWebフォームの例である。このWebフォームは、文字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6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るためである。

表6 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必須だが、手がかりがない。

図26 実験用 Web フォーム

6.2. 実験手順

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できる実験協力者は 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 をインストールしてもらい、使い方を把握してもらった。また、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協力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、実験用入力フォームで BADUI 改善タスクを行った。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。

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、ユーザの心的負荷の測定には日本語版 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 1 つに選択し、5 段階のリッカート尺度を用いる。

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、いらいら、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12段階のリッカート尺度を用いる。

6.3. 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 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 6 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 として検知されており、適用された変換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、不正解の変換フィルタが付与されたのは表 6 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、変換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 27, 28 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、変換フィルタが付与されるごとにエラー率を減少させる可能性が示唆された。

変換フィルタが付与されるごとの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った順番にアンケート結果のプロットを行

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 29 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 30 に示す. 図の縦軸と横軸は図 29 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.

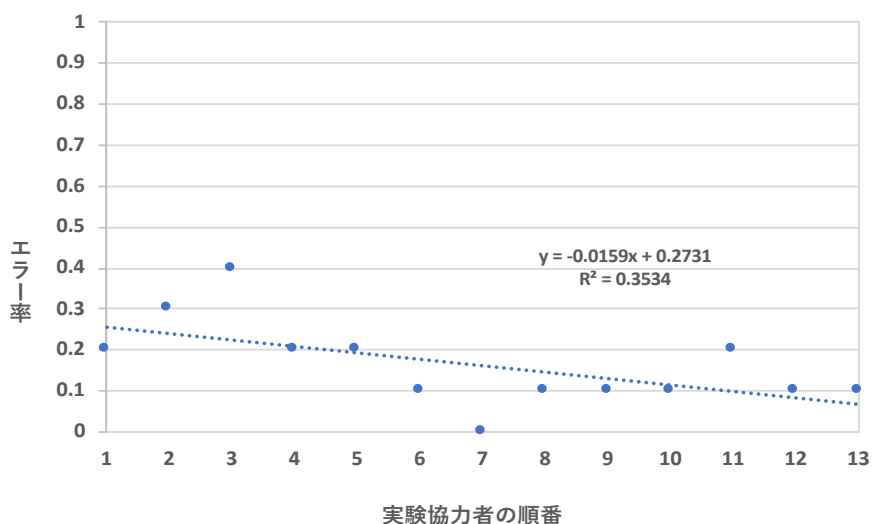


図 27 エラー率の推移 (BADUI ページのみ)

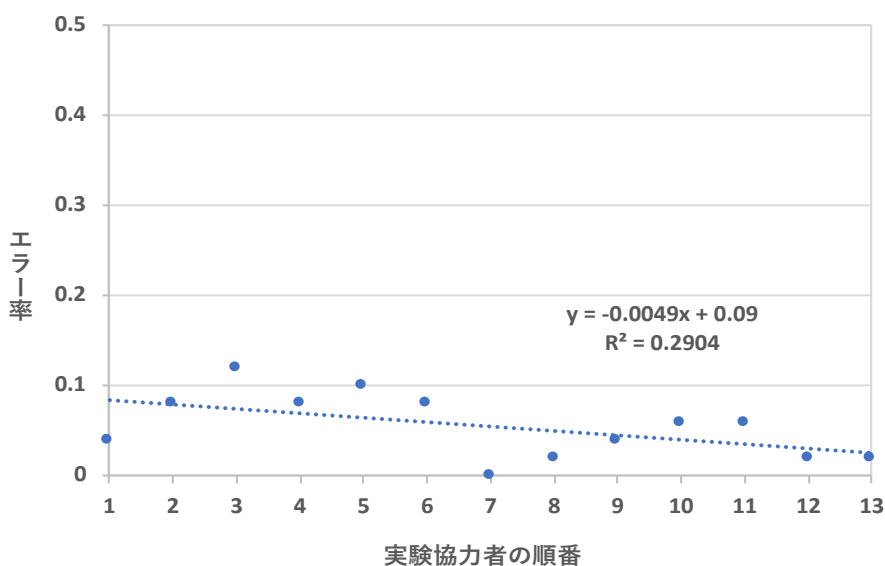


図 28 エラー率の推移 (全ページ)

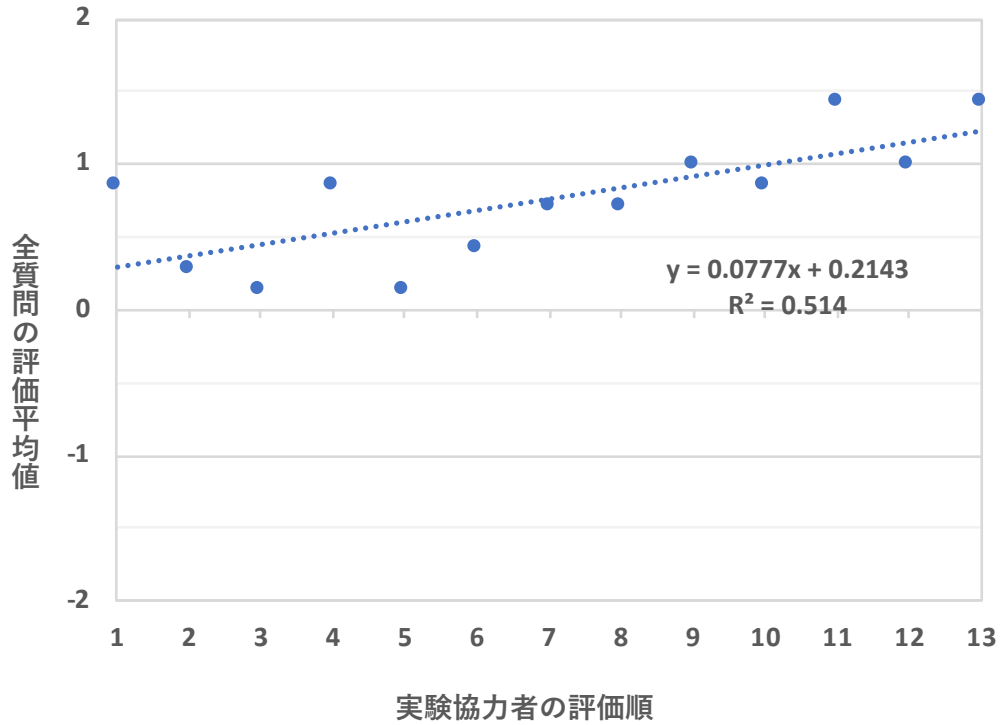


図 29 評価平均値の推移 (FUS)

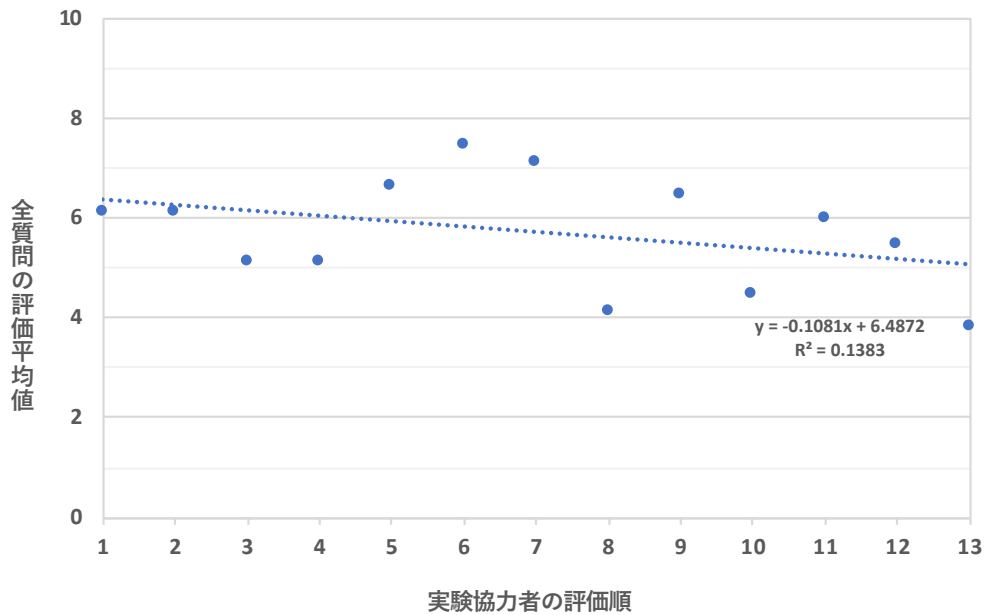


図 30 評価平均値の推移 (NASA-TLX)

6.4. 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 フォームを利用する度に 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可能性が示唆された。

しかし、本実験ではシステムを最初に入力ボックスへ付与したユーザの変換フィルタが反映され、上書きや取り消しが出来ないように設定していたため、そのユーザの付与した変換フィルタが不適切なものである場合を想定出来ていなかった。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、本実験の Web フォームは実験協力者ごとに異なる状況になっていることが多く、実験協力者の心理評価が偏ってしまっている可能性が考えられる。そのため、今後は全実験協力者に統一された実験環境でタスクを行ってもらふ必要があると考えられる。さらに、本実験では変換フィルタへの評価ボタンを実験協力者に利用するよう指示は出さなかったため、変換フィルタへの評価の様子は確認できなかったが、今後は肯定的な評価が多いフィルタについては、付与したユーザの信頼度を上げることや、否定的な評価が多いときに自動でフィルタを取り消す機能を追加することで、より不適切なフィルタの付与を防ぐことが可能になると考えられる。

ここで、システムが半分の BADUI 検知に失敗していたことから、WePatch 2 の BADUI 検知アルゴリズムの課題が明らかになった。その理由としては、例えば全角入力が必要な電話番号の入力項目において、ユーザが 09012341234 (不正解)、090-1234-1234 (不正解)、09012341234 (正解) の順番で試行錯誤した場合、本来「半角から全角に変換するフィルタ」が正解のところを、システムがハイフンを追加した時点で「電話番号にハイフンを補完するフィルタ」を推薦するためである (図 31)。

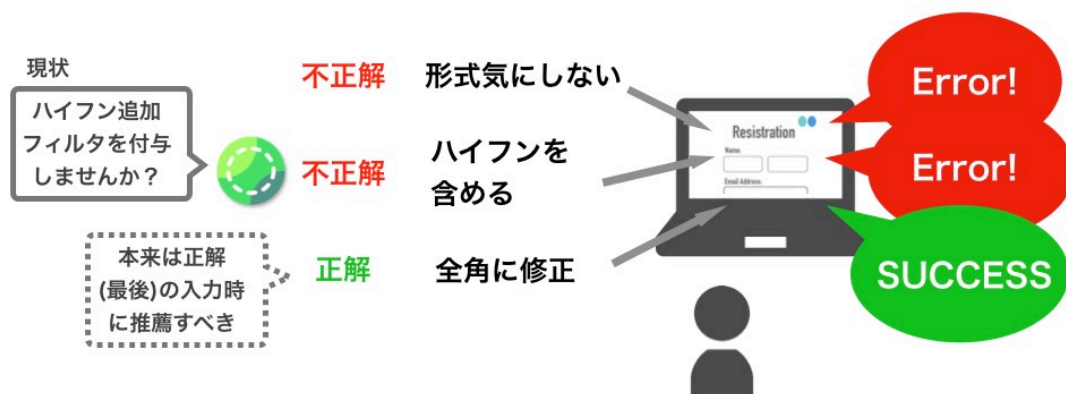


図 31 不正解の変換フィルタ推薦

これは、試行錯誤の中でも、何回目の入力された文字列の形式が正解の入力形式であるか、といったことを適切に考慮出来ていないためであると考えられる。そこで、正解の入力形式を考慮可能にすることにより、BADUI 検知精度の向上が可能であると考えられる。また、本実験では BADUI を含めた 10 ページ中 4 ページにおいて、全実験協力者が誤った入力をしなかったため、適切な BADUI の選定や、実験用フォームの設計が不適切であると考えられる。理由としては、実験用 Web フォームでは入力ボックスが 2, 3 個しか存在しなかったため、特に複数問の入力ボックスにおいてハイフンが必要であるかどうかの一貫性が欠如しているような BADUI において、ユーザの失敗を誘発しづらい状況になっていたと考えられる。そこで、実験用 Web フォームの難易度を高めることで、実際の BADUI の状況に近づけ、ユーザの失敗を誘発する状況にする必要があると考えられる。

最後に、本実験では WePatch 2 は何人程度のユーザが利用すれば有用なシステムであるか、どの程度 BADUI が改善可能であるかといった観点から有用性を評価出来ていなかった。そこで、次章では先述した BADUI 検知アルゴリズムと実験用 Web フォームの改良を行った上で、ページ内の BADUI を改善するまでに入力が必要な人数や検知精度を明らかにする。

第7章 BADUI 検知精度の評価実験

7.1. WePatch 2+

6章の実験の考察で述べた通り、WePatch 2ではBADUIをユーザの入力中に行なっていたため、ユーザの失敗の入力と正解の入力の区別がつかないことが原因で誤った変換フィルタを推薦してしまう問題があった。そこで、ユーザが入力を完了し、次のページへの遷移が成功するタイミングでBADUIの検知とフィルタ推薦を行う。遷移後に検知を行うことで、試行錯誤の最後の入力形式を正解の入力形式、それ以外を失敗の入力形式と区別することが可能である。なお、改良後のシステム名はWePatch 2+と呼ぶことにする。

実装に関しては、変換フィルタによる文字形式の補完や除去、データベースとの連携においては改良前のものと同様である。改良点としては、システムはユーザがページ遷移（リロードも含む）を行う度に入力された文字の形式情報の配列をウェブストレージへ追加したことである。それによって、ページ遷移後、ウェブストレージの最後に追加された配列を正解の入力操作、それ以外を失敗の入力操作と定義することが可能になり、最後に全ての失敗操作における入力形式情報と1回の正解操作における入力形式情報の比較を行う。WePatch 2+のBADUI検知と推薦プロセスと、ユーザに提示する推薦用のUIを図32、33に示す。図33のWePatchとタイトルが表記されている区画が推薦されたフィルタを適用するためのUIである。

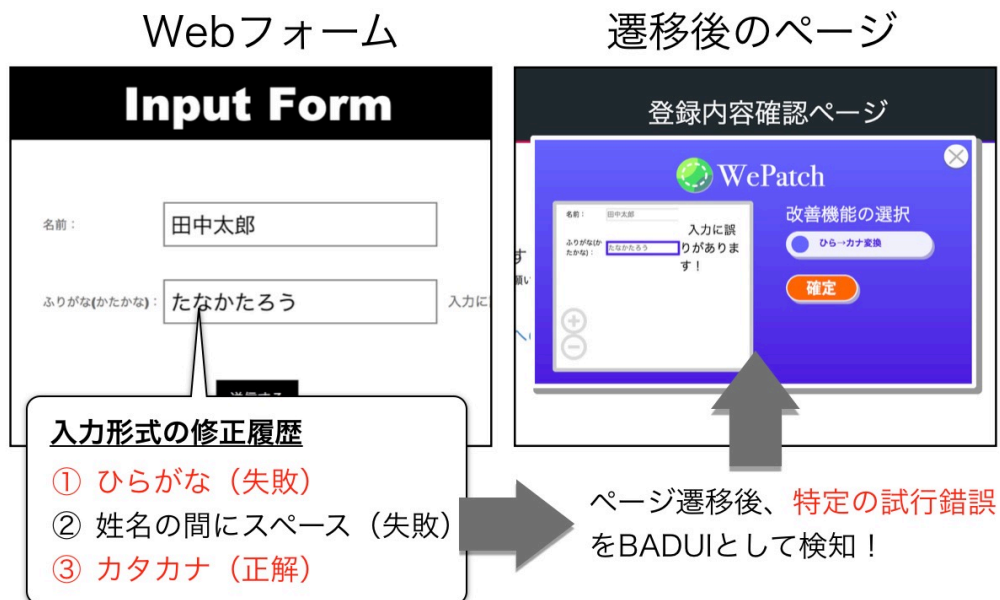


図32 変換フィルタ推薦の様子



図 33 変換フィルタ推薦の様子

その中でも左側のウィンドウから前ページの入力フォームを確認可能であり、右側で適用するフィルタの選択と送信を行うことが可能である。なお、推薦されたフィルタが表記されたボタンの上をユーザがマウスカーソルをホバーすると、適応対象の入力ボックスがハイライトするため、ユーザはどのフィルタがどの入力ボックスに適応されているかを把握可能である。また、フィルタ適応後の入力ボックスは緑色に変色するため、適応されたことが容易に把握可能である。

7.2. 実験準備

6章の評価実験における問題として、WePatch 2+はどの程度の人数が利用するとき、どの程度のBADUIが検知され、改善可能であるのかを検証できていないことが挙げられる。そこで、6章における実験用Webフォームの入力項目の少なさやBADUIの数の少なさとといった問題を考慮した上で、BADUIを含むWebフォームでの入力タスクを実施する。入力タスクを通して実験協力者が適応した変換フィルタや入力データから、どの程度の精度でWePatch 2+がBADUIを改善可能であるかを明らかにする。実験の準備として、まず実験協力者17名を募集した。実験協力者は10~20代の男性12名、女性5名であり、PCの使用頻度はほぼ毎日使用する実験協力者が15名、3日に1回程度使用する実験協力者が2名であった。タッチタイプを習得している実験協力者が1名、半分程度タッチタイプできる実験協力者が14名、キーボードを見ながら入力する実験協力者が2名であった。次に、入力タスクを行うための実験用フォームを用意した。この実験用Webフォームは6章の実験で用いたWebフォームの入力項目を2~5項目に増やした点以外は同様のものである。入力項目を増やした理由としては、前章で用いたWebフォームは入力項目が2~3個という

UI が単純すぎるものであり、実験協力者の BADUI における誤入力を誘発しづらい状況となってしまったためである。本実験では、この実験用 Web フォームのページを 70 ページ用意し、その内の 20 ページに BADUI が含まれるようにした。表 7 に用意した BADUI を示す。

表 7 実験用 Web フォームで用いた BADUI

分類		説明
ミスリード (単一)	A	ふりがなの手がかりはひらがな表記だがカタカナ入力が必要
	B	ふりがなの手がかりはカタカナ表記だがひらがな入力が必要
	C	電話番号の入力例では半角スペースが必要に見えるが、必要ない
	D	郵便番号の入力例はハイフンが必要な表記だが、ラベルの隣にはハイフン不要と書いてあり、実際はハイフン不要
ミスリード (複数間)	E	住所のみ全角入力だが他は全て半角入力
	F	年齢のみ全角入力だが他は全て半角入力
	G	誕生日のみ全角入力だが他は全て半角入力
	H	郵便番号はハイフン必要ではないが、電話番号は必要
手がかりの欠如 (単一)	I	フリガナをカタカナで入力する必要があるが、入力例やラベルといった手がかりが欠如している
	J	郵便番号でハイフン必要だが表記されていない
	K	生年月日でスラッシュ必要だが表記されていない
	L	電話番号でハイフン必要だが表記されていない
慣習との乖離 (単一)	M	姓名の間にスペースが必要
	N	誕生日がスラッシュでなくハイフンで区切らなければいけない
	O	電話番号を国番号から記述する必要がある
	P	郵便番号をスラッシュ区切りで記述しなければいけない
一苦労 (単一)	Q	名前で半角カタカナ入力が必要
	R	名前英字を全角英字入力が必要
	S	メールアドレスで全角英字入力が必要
	T	住所フリガナで半角カタカナ入力が必要

7.3. 実験手順

実験を行った手順について記述する。まず、入力準備として普段から利用している PC に WePatch 2+ をインストールしてもらった上で、実験中に WePatch 2+ 以外の拡張機能はオフの状態にしてもらった。また、WePatch 2+ の使い方を説明することで BADUI 改善方法について把握してもらった。さらに、ダミープロフィールの入力に慣れてもらうため、練習用の Web フォームでそれを 3 回入力してもらった。入力タスクでは実験協力者にダミープロフィールを参考にしながら、提示される複数の Web フォームで入力するよう依頼した。その際、もし WePatch 2+ から変換フィルタを推薦された場合、適切だと思われるフィルタを付与してもらった。最後に、入力タスク終了後は実験協力者のプロフィールやシステムのフィードバックを得るためのアンケートに回答してもらった。

ここで、実験を通して得られるデータを以下に記述する。

- **ユーザの入力データ**：ユーザが入力した文字列、また、名前や郵便番号といった入力された項目名
- **推薦フィルタデータ**：WePatch 2+ が BADUI を検知し、その際に推薦された変換フィルタ名、また、それが推薦された入力項目名
- **適応済みフィルタデータ**：WePatch 2+ から推薦されたフィルタの中でもユーザによって適応が行われた変換フィルタ名、また、それが推薦された入力項目名
- **フィードバック**（使いやすかった点 / 使いづらかった点について自由記述）

7.4. 評価指標

実験ではクラスは 2 つあり、BADUI であるクラス（正例）と非 BADUI（負例）である。この 2 クラスにおける評価尺度として適合率（Precision）と再現率（Recall）を用いる。ここで、クラス C_i についての適合率と再現率は以下のように算出される。以下の TB は入力（テキスト）ボックスを表す。

$$\text{Precision}(C_i) = \frac{\text{正しく } C_i \text{ が改善された TB 数}}{C_i \text{ として改善された TB 数}}$$

$$\text{Recall}(C_i) = \frac{\text{正しく } C_i \text{ が改善された TB 数}}{C_i \text{ に属する TB 数}}$$

正しく改善された基準は、誤入力を誘発する原因が解消される変換フィルタが付与されている場合とした。例えば、「ふりがなの手がかりはひらがな表記だがカタカナ入力が必要」

の BADUI であれば、ユーザはカタカナ形式で入力しなければいけないところを、誤ってひらがなで入力してしまうと考えられる。この場合の正解フィルタは「ひらがなからカタカナへ変換」のフィルタを正解とする。また、BADUI の入力ボックスに正解のフィルタに加え、正解フィルタ以外のフィルタが付与されている場合であっても Web フォームを改悪してしまうフィルタでない限り正解とした。

ここで、入力タスクにおいて、システムが BADUI を正しく検知したにも関わらず、ユーザが推薦されたフィルタは BADUI を改善できないと判断してフィルタの適応を行わなかった場合や、フィルタの適応忘れによって正解のフィルタの適応漏れが起きてしまう可能性が考えられる。また、ユーザの個人情報を入力した後にシステムが何かしらのデータをサーバに送信すると抵抗を感じるユーザが多いと考えられるため、自動検知はあくまで理想的な仕組みとしてのものを意味している。そこで、上記の改善された BADUI についての適合率と再現率とは別に、ユーザが関与しない部分である、システムが検知した BADUI についての適合率と再現率も算出する。システムが検知した BADUI についての適合率と再現率は以下のように算出する。

$$\text{Precision}(C_i) = \frac{\text{正しく } C_i \text{ と検知された } TB \text{ 数}}{C_i \text{ として検知された } TB \text{ 数}}$$

$$\text{Recall}(C_i) = \frac{\text{正しく } C_i \text{ と検知された } TB \text{ 数}}{C_i \text{ に属する } TB \text{ 数}}$$

7.5 節以降では、システムから推薦された変換フィルタをユーザが改善を行うことを半自動改善、また、システムがユーザの試行錯誤情報から特定の BADUI であると検知することを自動検知と呼ぶことにする。

7.5. 結果

用意した BADUI の妥当性を検証するため、実験協力者の誤った操作に着目したものを表 8 に示す。表 8 の 1 行目の BADUI 分類とアルファベットは表 7 に示した BADUI と対応しており、エラー回数はユーザが誤った入力を行った平均回数、誤答率はユーザが誤った入力を 1 回以上した平均回数を表している。また、表 8 の中でも BADUI の Q はユーザの誤答率が 0%であったため、不適切な BADUI であるとし、以降の結果算出の過程で BADUI の Q は考慮しない。まず、表 8 の誤答率を全て平均すると 49.12%となり、用意した BADUI のページにおいて実験協力者のほぼ半分が間違えた入力をしたことがわかる。そのため、本実験では半分程度のユーザが失敗してしまう BADUI を設定できていたと言える。

次に、自動検知、または半自動改善が成功していたかどうかについて、各 BADUI 分類についてまとめたグラフを図 34 に示す。図 34 の横軸は各 BADUI 分類、縦軸は BADUI の再現率を表している。また、青色のグラフは半自動改善、オレンジ色のグラフは自動検知が適切に行われた BADUI の再現率の平均を表している。表 8 と図 34 より、BADUI 分類の中でも誤答率が高かった「ミスリード (単一)」と「慣習との乖離 (単一)」の BADUI 改善または検知はしやすく、一方で、誤答率の低かった「ミスリード (複数)」の BADUI 改善または検知は難しいことがわかる。

表 8 各 BADUI におけるユーザの入力試行錯誤データ

分類		エラー回数	平均	誤答率 (%)	平均 d (%)
ミスリード (単一)	A	1.00	1.43	41.2	58.8
	B	1.00		58.8	
	C	2.71		100.0	
	D	1.00		35.3	
ミスリード (複数間)	E	1.00	1.07	64.7	33.8
	F	1.29		41.2	
	G	1.00		5.9	
	H	1.00		23.5	
慣習との乖 離 (単一)	I	1.00	7.91	82.4	66.2
	J	3.25		47.1	
	K	26.38		94.1	
	L	1.00		41.2	
一苦勞 (単一)	M	1.50	2.32	11.8	51.5
	N	3.81		94.1	
	O	2.64		64.7	
	P	1.33		35.3	
手がかりの 欠如 (単一)	Q	0.00	1.19	0.00	35.3
	R	1.00		29.4	
	S	1.57		41.2	
	T	1.00		35.3	

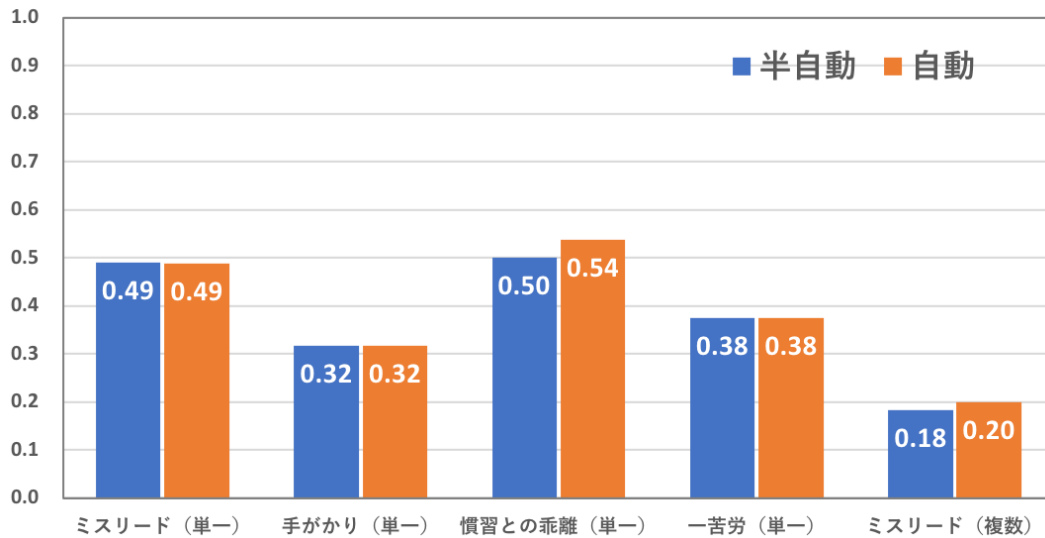


図 34 BADUI 分類ごとの改善・検知率

さらに、半自動改善を行った BADUI が正解かどうかを、7.4 節で記述した基準に則り人手で分類した。その結果として得られた人数ごとの平均再現率の遷移を表すグラフを図 35, 36 に示す。横軸は実験協力者の人数であり、縦軸は実験協力者数ごとにその人数の全ての組み合わせから得られる再現率を平均化したものを表している。図 35, 36 より、1 人が利用するときの再現率は約 4 割と低い結果であるが、5 人が利用した時点で約 8 割、10 人時点では約 9 割に達していた。また、実験協力者 17 名全員による BADUI の適合率は半自動改善が 0.75、自動検知では 0.64 であり、半自動改善の方が BADUI の誤検知の可能性は低いことが明らかになった。さらに、実験協力者全員により半自動改善または自動検知されなかった BADUI はどちらも 19 件中 1 件であった。これは、BADUI の入力ボックスにおいて誤入力は起きていたが、異なる形式が混在する文字列から 1 つの形式の文字列へのユーザの修正行動（中野区中野 4-15（半角と全角形式の混在）から中野区中野 4-15（全角形式のみ）へ修正等）をシステムが BADUI として検知出来ていないことが原因であった。最後に、実験終了後に収集したシステムについてのフィードバックについて記述する。まず、システムの使いやすかった点、または分かりやすかった点の質問項目では、「自分が何の入力を間違えたのかを提示してくれるので、ここを直してほしいとすぐに選べる点は使いやすかったです」「ポインターを乗っているときにどこのフォームを改善しようとしているのか左の画面で示されるのが良かった」といったような、どの入力ボックスが BADUI として検知され、修正すると良いかを把握しやすかった点や、「ボタンを押すだけでいいので便利」「ワンクリックで選択できる点」といったような改善の手間が少ないことを示唆するような意見が得られた。システムの使いにくかった点、または分かりづらかった点の質問項目では、「左画面内の文字が小さく少し見えづらかった」「改善すべきページが縮小されていて文字が小さく、改行も多かったので見づらい」といったように前ページの Web フォームを

再現した箇所が見えづらいといった意見が得られた。また、「どのように修正されるかを左でデモして欲しかった」「補完の前後での変化が確認できたらいいと思った」といったような、適応するフィルタによって、どのような文字修正が行われるかが把握しにくいといった意見が得られた。

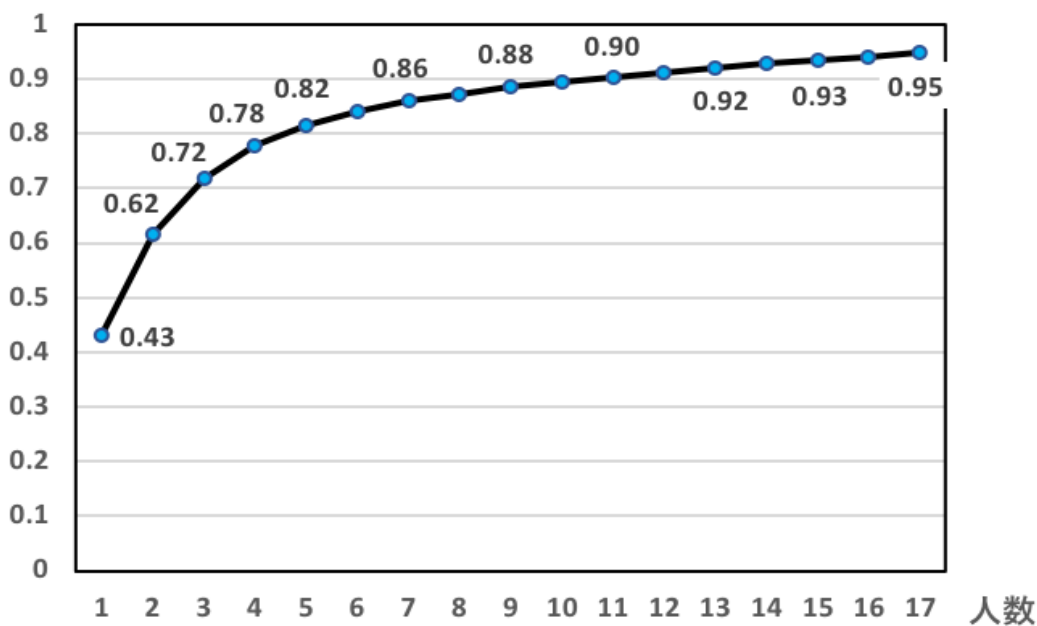


図 35 人数ごとの BADUI 平均再現率の遷移（半自動）

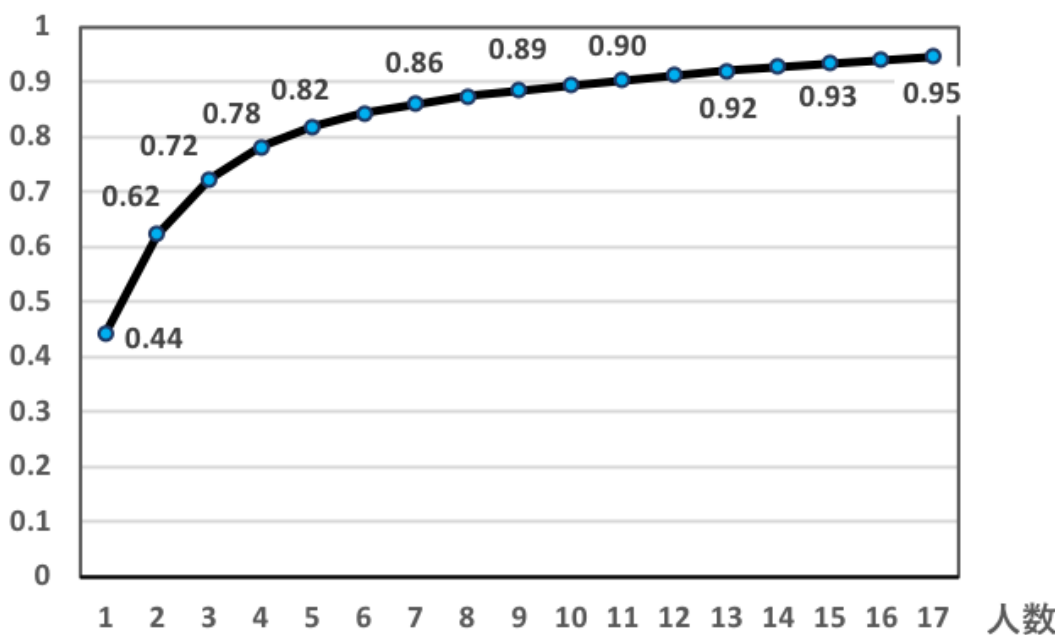


図 36 人数ごとの BADUI 平均再現率の遷移（自動検知）

7.6. 考察

実験結果より、システムは少数のユーザの手で多くの BADUI が改善可能であることが明らかになった。図 34 より、特に Web フォームの BADUI の中でも「ミスリード (単一)」と「慣習との乖離 (単一)」の BADUI に対して有用であることが明らかになった。これらの BADUI では、ユーザが特に誤った入力をしてしまいやすいものであるため、そういった致命的な UI における誤操作を防ぎやすい点は有用であると考えられる。一方で、「ミスリード (複数)」の BADUI に対しては有用でないことが結果から明らかになった。これは、こういった一貫性のない BADUI は、ユーザは形式を使い分ける必要がある点で使いづらいが、誤って入力してしまうほどの要因になりづらく、BADUI の検知が起きづらい原因になったと考えられる。また、自動検知の方がユーザ 1 人による改善率が高い BADUI 分類があったが、ユーザのフィードバックにあった「改善対象のフォームのレイアウトが小さくて見づらい」や「どのように文字が修正されるのかが分からない」といったコメントから、WePatch 2+のフィルタ推薦画面のレイアウトに問題があったため、修正動作を行うことに失敗し、「慣習との乖離 (単一)」や「ミスリード (複数)」において、自動検知の方がやや半自動改善の結果を上回る原因となったと考えられる。さらに、BADUI でない入力ボックスにフィルタを付与してしまった理由の 1 つも同様の原因によるものと考えられる。

ここで、ユーザの手を介さずとも、自動的にシステムが変換フィルタを適用する自動 BADUI 改善手法も考えられるが、図 34, 35, 36 から半自動改善と自動検知の結果に差があまり見られなかったため、半自動な改善であっても問題なく BADUI を改善できるものである。また、適合率が半自動改善のほうが高い結果であった。さらに自動検知を採用する場合、サーバにどのような情報が送られているのか心配である、といったユーザの不安を減らすことができるものであるため、結論としては半自動 BADUI 改善手法の方が有用であると考えられる。なお、適合率の結果から BADUI でない入力ボックスを誤って BADUI と検知され、変換フィルタが推薦されてしまったものが数件あった。検知されたフィルタデータを確認したところ、いずれの推薦された変換フィルタはユーザの誤入力を増やしてしまうようなものではなかった。これは、WePatch 2+は正解の入力形式の考慮が可能であり、内部のプログラムとデータベースが求める入力形式に則った変換フィルタの推薦を行うことが可能になったためであると考えられる。そのため、誤検知した場合であっても、ユーザビリティに悪影響を与える変換フィルタが推薦される可能性は少ないと考えられる。しかし、偶然全ての変換フィルタが改悪しないものであった可能性は排除しきれないため、今後も変換フィルタの信頼性の評価を行う機能は必要であると考えられる。

本章では表 1 の Web フォームの BADUI の 8 分類のうち、5 種類のみ扱って評価実験を行ったが、今後は「手がかりの欠如 (複数)」「慣習との乖離 (複数)」の BADUI についても、改善機能を実現可能であると考えられる。具体的には、複数の入力ボックスに対して住所 1、住所 2 しか手がかりがないために、どこで区切ればいいのか分かりづらい BADUI で

は、正例のユーザ群の入力した文字数や入力形式以外の文字列に対してアラートを表示するといった支援方法が考えられる。また、今回用意した手法は、2章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。また、現状の WePatch 2+ では1つの入力ボックスに既に変換フィルタが付与されている場合、他のユーザによって変換フィルタを追加することができないものであった。つまり、BADUI 1 つにつきユーザの間違え方は1つしかないという前提の改善方法になっており、人によって間違え方が異なる可能性を考慮できていなかった。そのため、複数の失敗パターンの修正を可能にする改良を行う必要があると考えられる。

第8章 おわりに

8.1. まとめ

本稿では、まず、WePatch の入力 of 自動修正や手がかり提示による改善機能が BADUI におけるユーザの操作の負担を減らせるかどうか検証した。その結果として、入力の自動修正を行う変換フィルタ機能はユーザの負担を減らすことが可能であることが明らかになった。一方で、過去の多くの研究で提案されてきた、ユーザの手によって提示される付箋のアノテーション機能が逆にユーザの負担を増やしてしまうことを明らかにした。次に、従来のユーザの手による BADUI 改善手法における手間を減らすため、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、また、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。さらに、提案手法を実現するため、変換フィルタの半自動化を実現する WePatch 2 を実装した。さらに、正解の入力形式を考慮可能することで WePatch 2 を改良した WePatch 2+ の BADUI 検知精度の評価実験により、5 人以上が BADUI の入力ボックスで入力することにより有用であること、また、半自動 BADUI 改善手法の方が自動 BADUI 改善手法より有用であることを明らかにした。

8.2. リミテーション

半自動 BADUI 改善手法はユーザの失敗の操作と正解の操作を利用する手法であるため、ユーザにとって悩ましいが失敗しづらい UI は改善しづらい。理由としては、多くのユーザにとって使いづらい UI であったとしても、使いづらい点に気づきやすく対処可能な BADUI であれば失敗が生じにくいいため、BADUI の改善まで多くのユーザを要すると考えられる。次に、WePatch 2+ の実装上の都合によるリミテーションも存在する。具体的には、Web ページによっては URL の遷移が想定外のものがあり、BADUI 検知が不可能である。また、7.5 節で記述したように特定の入力形式が混在してしまう場合、BADUI の検知が不可能である場合がある。さらに、表 1 の BADUI のうち改善対象の 5 分類と、将来的に改善可能になると考えられる「手がかりの欠如 (複数)」「慣習との乖離 (複数)」を除いた「一苦勞 (複数)」の BADUI については試行錯誤が起きづらいため改善が困難であると考えられる。

8.3. 今後の応用先

今後の応用先としては、本システムを通してユーザの手で変換フィルタが付与されることによって、膨大な数の BADUI ページを収集可能であると考えられる。そして、これまでに無いような情報量の BADUI ページが蓄積された BADUI データセットを構築することが可能であると考えられる。現在の WePatch 2+ では半自動 BADUI 改善手法の方が有用であると結論づけているが、はじめは半自動改善手法をユーザに利用してもらうことで BADUI データセット構築し、機械学習を用いることで Web ページ情報からの BADUI 検

知の可能性を検証する。そして、BADUI 検知精度を高めることで、将来的に自動 BADUI 改善手法の実現が期待できる。

謝辞

Web 公開版からは謝辞を削除しました

参考文献

- [1] Downes, P. K.: Creating a practice website. *British Dental Journal*, Vol.202, pp.597-604 (2007).
- [2] Wroblewski, L.: *Web Form Design: filling in the blanks*, Rosenfeld Media (2008).
- [3] Nielsen, J. and Loranger, H.: *Prioritizing Web Usability*, Pearson Education (2006).
- [4] Gregg, D.G. and Walczak, S.: The relationship between website quality, trust and price premiums at online auctions, *Electronic Commerce Research*, Vol.10, pp.1-25 (2010).
- [5] 中村聡史: 失敗から学ぶユーザインタフェース, 技術評論社 (2015).
- [6] Ranganathan, C. and Ganapathy, S.: Key dimensions of business-to-consumer Web sites, *Information Management*, Vol.39, pp.457-465 (2002).
- [7] 10 Usability Heuristics for User Interface Design, <https://www.nngroup.com/articles/ten-usability-heuristics/> (参照 2018 月 07 日 25 閲覧).
- [8] Bailey, R.W. and Koyani, S.J.: *Research-Based Web Design & Usability Guidelines* (2005).
- [9] IBM Lotus User Interface Developer Documentation, <http://www-12.lotus.com/ldd/doc/oneuidoc/docpublic/guidelines/coding.htm>, (参照 2018 月 12 日 20 閲覧).
- [10] Apple Web Design Guide, <http://www.idemployee.id.tue.nl/g.w.m.rauterberg/publications/AppleWDG96.pdf>, (参照 2018 年 12 月 20 日閲覧).
- [11] Avila, A., Andrés, J., Sandra, R., Tuch, A.N., Sébastien, O. and Klaus, O.: Simple but Crucial User Interfaces in the World Wide Web: Introducing 20 Guidelines for Usable Web Form Design, Intech (2010).
- [12] Seckler, M., Heinz, S., Bargas-Avila, J.A., Opwis, K. and Tuch, A.N.: Designing usable web forms, *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pp.1275-1284 (2014).
- [13] Seckler, M., Heinz, S., Bargas-Avila, J.A., Opwis, K. and Tuch, A.N.: Empirical evaluation of 20 web form optimization guidelines, In *Proc. of CHI EA '13*, pp.1893-1898 (2013).
- [14] Nielsen, J.: Heuristic evaluation of user interfaces, In *Proc. of CHI'90*, pp.249-256 (1990).
- [15] Nielsen, J.: Usability Inspection Methods, In *Proc. of CHI '94*, pp.413-414 (1994).
- [16] Brooke, J.: SUS: a “quick and dirty” usability scale, *Usability Evaluation in Industry*, pp.189-194 (1996).
- [17] Sebastien, O., Sc, M., Prof Dr., Opwis, K. and Aeberhard, Andreas.: FUS - Form Usability Scale Development of a Usability Measuring Tool for Online Forms, Unpublished master's thesis (2011).

- [18] Chiew, T.K., and Salim, S. S.: Webuse: Website usability evaluation tool, *Malaysian Journal of Computer Science*, Vol.16, No.1, pp.47-57 (2003).
- [19] UX CHECK, <https://www.uxcheck.co/>, (参照 2018 年 12 月 20 日閲覧).
- [20] Bakaev, M., Mamysheva, T. and Gaedke, M.: Current trends in automating usability evaluation of websites: can you manage what you can't measure?, In *Proc. of IEEE 11th International Forum on Strategic Technology (IFOST)*, pp.510–514 (2016).
- [21] Chi., E.H.: Improving Web Usability Through Visualization, *IEEE Internet Computing*, Vol.6, pp.64-71 (2002).
- [22] Brainerd, J. and Becker, B.: Case Study: E-Commerce Clickstream Visualization, In *Proc. of IEEE '01. Information Visualization*, pp.153-158 (2001).
- [23] Cadez, I., Heckerman, D., Meek, C., Smyth, P. and White, S.: Visualization of Navigation Patterns on a Web Site Using Model-Based Clustering, In *Proc. KDD '00*, pp.280-284 (2000).
- [24] Herder, E. and Weinreich, H.: Interactive Web Usage Mining with the Navigation Visualizer, In *Proc. CHI '05*, pp.1451-1454 (2005).
- [25] Vigo, M. and Harper, S.: Real-time detection of navigation problems on the World 'Wild' Web, *International Journal of Human-Computer Studies*, Vol.101, pp.1-9 (2017).
- [26] Bakaev, M., Heil, S., Khvorostov, V. and Gaedke, M.: HCI Vision for Automated Analysis and Mining of Web User Interfaces, In *Proc. ICWE '18*, Vol.10845, pp.136-144 (2018).
- [27] Grigera, J., Garrido, A., Rivero, J. M. and Rossi, G.: Automatic detection of usability smells in web applications. *International Journal of Human-Computer Studies*, Vol.97, p.129-148 (2017).
- [28] Santana, V.F.D. and Baranauskas, M.C.C.: WELFIT: A remote evaluation tool for identifying Web usage patterns through client-side logging, *International Journal of Human-Computer Studies*, Vol.76, pp.40-49 (2014).
- [29] Paternò, F., Schiavone, A.G. and Conti, A.: Customizable Automatic Detection of Bad Usability Smells in Mobile Accessed Web Applications, In *Proc. MobileHCI '17*, No. 42 (2017).
- [30] Nebeling, M., Speicher, M. and Norrie, M.: W3touch: Metrics-based Web Page Adaptation for Touch, In *Proc. CHI '13*, pp.2311-2320 (2013).
- [31] Hartson, H.R. and Castillo, J.C.: Remote evaluation for post-deployment usability improvement, In *Proc. AVI '98*, pp.22–29 (1998).
- [32] Zarraonandia, T., Díaz, P., Aedo, I. and Montero, A.: Augmentation as a Promising Technology for End User Development, In *Proc. AVI '16*, pp.346-347 (2017).
- [33] Reed, D. and John, S.: Web annotator, In *Proc. SIGCSE '03*, Vol.35, pp.386-390 (2003).
- [34] Chen, Y.C., Hwang, R.H. and Wang, C.Y.: Development and evaluation of a Web 2.0 annotation system as a learning tool in an e-learning environment. *Computers & Education*,

Vol.58, pp.1094–1105 (2012).

- [35] Tajima,K. and Nakamura, S.: WePatch: A System Enabling Users to Improve Bad User Interfaces on the Web, In *Proc. OZCHI'17*, pp.448-451 (2017).
- [36] Dong, T., Ackerman, M. S., Newman, M.W. and Paruthi, G.: Socialoverlays: Collectively making websites more usable, In *Proc. INTERACT '13*, Vol.8120, pp. 280–297 (2013).
- [37] J.Kahan.,M.-R.Koivunen.,E.Prud'Hommeaux.,R.R.Swick.: Annotea: an open RDF infrastructure for shared Web annotations, In *Proc. WWW '01*, Vol.39, pp.623-632 (2001).
- [38] txtpen, <https://betalist.com/startups/txtpen>, (参照 2018 年 12 月 06 日).
- [39] Hypothesis, <https://web.hypothes.is/>, (参照 2018 年 12 月 06 日).
- [40] Yee, K.P.: CritLink: advanced hyper links enable public annotation on the Web, In *Proc. CSCW '02*(2002).
- [41] Sergio,T,S. and Ackerman,I, I.. Enabling communication between users surfing the same web page. 2003.
- [42] Swearngin, A., Ko, A.J. and Fogarty, J.: Genie: Input Retargeting on the Web through Command Reverse Engineering, In *Proc. CHI '17*, pp.4703-4714 (2017).
- [43] Yao, Z. and Sen, A.: Implementation of the AutoComplete Feature of the Textbox Based on Ajax and Web Service, *JOURNAL OF COMPUTERS*, Vol.8, No.9, pp.2197-2203 (2013).
- [44] Trinh, T.D., Wetz, P., Do, B.L., Aryan, P.R., Kiesling, E. and Tjoa, A.M.: An Autocomplete Input Box for Semantic Annotation on the Web. In *Proc. ISWC '15*, Vol.1456, pp.97-103 (2015).
- [45] BADUI タレこみサイト, <http://up.badui.org/>,(参照 2018 年 12 月 21 日参照)

業績

登壇発表

- [1] 田島一樹, 中村聡史. ストーリーコンテンツに対するネタバレの基礎調査とその判定手法の検討, 研究報告グループウェアとネットワークサービス (GN), Vol.2015-GN-96, Issue.7, pp.1-6, 2015.
- [2] 田島一樹, 中村聡史. Twitter におけるアニメのネタバレツイート判定手法の提案, 第8回データ工学と情報マネジメントに関するフォーラム (DEIM2016), Vol.B5-4, pp.1-8, 2016.
- [3] 田島一樹, 中村聡史. Web 上の BADUI をユーザの手で改善可能とするシステムの提案, 第24回インタラクティブシステムとソフトウェアに関するワークショップ (WISS2016) 予稿集, 2016.
- [4] 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, 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.2017-HCI-172, Issue.23, pp.1-8, 2017.
- [5] 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) ,2017-HCI-174(16),1-7 (2017-08-16) , 2188-8760.
- [6] Kazuki Tajima, Satoshi Nakamura.2a WePatch: A System Enabling Users to Improve Bad User Interfaces on the Web, OzCHI 2017, pp. 448-451, November 2017, Brisbane, QLD, Australia.
- [7] 久保田夏美, 田島一樹, 松田滉平, 中村聡史. リアルタイム感情記録による日記習慣化支援手法の検討.信学技報, vol. 118, no. 49, HCS2018-26, pp. 165-170.
- [8] 田島一樹, 中村聡史, 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 , 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) , 2018-HCI-179(3), 1-8 (2018-08-13) , 2188-8760.
- [9] 田島一樹, 中村聡史: WePatch2: 無数のユーザによる半自動的な Web フォームの BADUI 改善システム, 情報処理学会 研究報告グループウェアとネットワークサービス (GN) , 2019-GN-106 (19), 1-8 (2019-01-17), 2188-8744 (2019/01/24).

共著

- [1] 田村柁優紀, 新納真次郎, 白鳥裕士, 田島一樹, 中村聡史. Uniotto: グループ型音楽鑑賞手法の提案と実装, ワークショップ2017 (GN Workshop 2017) 論文集,2017,1-8 (2017-11-09).
- [2] 水野颯, 田島一樹, 牧良樹, 中村聡史: 放送コンテンツに同期したツイートをを用いたネ

タバレシオン推定手法の検討, 情報処理学会 究報告グループウェアとネットワークサービス (GN) , 2018-GN-105(11),1-8 (2018-05-03) , 2188-8744. (2018/05/11).

- [3] 松山直人, 田島一樹, 中村聡史, ユーザの視線情報分析に基づく現実空間の BADUI 検出に関する検討, 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) , 2018-HCI-180, 1-8, (2018-11-27), 2188-8760.
- [4] 佐々木美香子, 田島一樹, 神山拓史, 中村聡史: ライフログ写真の着衣色変化によるファッションへの意識変化手法の提案, 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) , 2019-HCI-181 (6), 1-8 (2019-01-14), 2188-8760 (2019/01/21).

受賞

- [1] プレゼンテーション賞 (第 8 回データ工学と情報マネジメントに関するフォーラム)
- [2] Special Innovation Award in collaboration with Ash Center, HKS, Harvard (チャレンジ!! オープンガバナンス)
- [3] GN 研究会優秀発表賞 (第 106 回グループウェアとネットワークサービス研究会)