WePatch: A System Enabling Users to Improve Bad User Interfaces on the Web*

Kazuki Tajima
University of Meiji
Nakano 4-21-1, Nakano, Tokyo, Japan
kazu.t.1205@gmail.com

Satoshi Nakamura
University of Meiji
Nakano 4-21-1, Nakano, Tokyo, Japan
malago@affiliation.org

ABSTRACT

Many[1] user interfaces on the Web are confusing for users to operate or make users more prone to mistakes, and we call such a user interface a "BADUI" (BAD User Interface). To solve these problems, we proposed WePatch, a system enables users to improve BADUIs on the Web by attaching improvement functions virtually. We implemented a prototype WePatch system as a browser extension that had functions for correcting inputted text automatically, and adding annotations and so on. Also, we experimentally compared the usability of BADUIs before and after being qualitatively improved by WePatch. The results revealed that WePatch considerably improved the usability of BADUIs.

CCS CONCEPTS

• **Human-centered systems computing → User interface design**;

KEYWORDS

Usability, user interface, Web service, annotation, BADUI

ACM Reference format:
Kazuki Tajima, Satoshi Nakamura. 2017. WePatch: A System Enabling Users to Improve Bad User Interfaces on the Web. In *Proceedings of the 29th Australian Conference on Human-Computer Interaction, Brisbane, QLD, Australia, November 2017 (OzCHI 2017)*, 4 pages. https://doi.org/0000001.0000001

1 INTRODUCTION

With the spread of the Internet, the number of Web sites now exceeds one billion (see https://news.netcraft.com/archives/2016/03/18/march-2016-web-server-survey.html). Such sites provide a wide variety of services, including online shopping, online reservations, and communication on electronic bulletin boards, and together represent a vast collection of information and knowledge. These services have made it possible for users to search for necessary information, purchase desired products, or communicate with others while staying at home. As a result, they have become indispensable for users. Thus, Web sites need to have a good user interface that enables users to accomplish their objectives efficiently. However, depending on their design and engineering, user interfaces may become unintentionally complicated to use. We call such a user interface a "BADUI (Bad User Interface)" [9], and users have posted many BADUI examples on Twitter with the hash-tag #BADUI. Here, BADUIs on the Web are classified based on the basic five problems:

- **Lack of clues:** A Web page does not provide sufficient information on what to input and how to input it.
- **Non-user-friendly feedback:** A Web page returns technically complicated or slowly feedback to users.
- **Inconsistency:** If a user encounters a Web page containing two form fields, such as one that requires slashes and another that does not, s/he is probably not going to be able to use the required format.
- **Lack of Maintenance:** A Web page displays outdated information that may confuse users.
- **Deception:** Several Web pages show pre-checked checkboxes at the bottom of pages that users may be unaware of to register for e-mail services.

BADUIs may cause users to perform wrong operations and may increase subjective stress. There have been cases where users were provided undesired additional services after performing erroneous operations, and the resultant stress made them abandon the desired main services. Therefore, many methods have been proposed for evaluating the usability of Web sites and supporting UI improvements [4,5]. However, even if developers use an evaluation method, it is not uncommon for users to find a BADUI after checking out the Web site. Although the problem can be solved by requesting improvement modifications, the improvement process is expensive.

*Produces the permission block, and copyright information
[1]It is a datatype.

OzCHI 2017, November 2017, Brisbane, QLD, Australia

DOI: 0000001.0000001

Consequently, Web sites are not made easier, and users have to keep using a hard-to-use Web site. In this paper, we propose a method that enables ordinary users to solve the problems of BADUIs on Web sites by attaching a function to an input form such as a conversion filter, attaching an annotation, and sharing addressed Web sites with others. Also, we implement a prototype system based on our method. Then, we evaluate the proposed system and verify its usefulness by conducting an experimental test.

The contributions of our work are to realize a system that enables users to improve BADUIs on the Web virtually and easily, and to clarify the usefulness and characteristics of our system by conducting an experimental test.

2 RELATED RESEARCH

Battleson et al. showed that an enormous burden was imposed on users when searching for necessary information on the Web. Therefore, they proposed a usability test to evaluate concepts such as ease of use to search [1]. Palmer identified appropriate criteria for measuring usability by examining what kinds of designs make a good impression on user or are so complicated to use that they can be termed "fatal" in a business sense [5]. Also, Hong et al. developed tools to record the usage situation efficiently and to analyze and visualize screen captures to support the usability tests of Web sites [4]. Such usability evaluation methods are useful for discovering the deficiencies of Web sites. However, they do not always find such deficiencies. Also, because it is developers who determine UIs on Web sites, the "deception" BADUIs introduced in Chapter 1 may not be rectified. On the other hand, our system can improve BADUIs that have passed a usability test and "deception" BADUIs by calling the users' attention to them so that they can share the information among themselves.

Considerable research has been done to allow users to add annotations to Web sites for various purposes. Annotea allows users to annotate any Web content using an RDF (resource description framework) and share it with other users [6]. CritLink can add a hyperlink to a string on a Web page and display it without requiring system installation on any browser [7]. Reed and John developed a system that allows users to add and share annotations like the "Web advisory sticker" function of our system that can add label stickers [2]. However, one of the things our Web advisory sticker aims to improve is BADUIs, and our system can improve various BADUIs by combining multiple functions. An attractive system for users has been developed that allows them to communicate with other users in the same way as a Web chat [10]. The system makes it possible for users to communicate in real time, so they can ask other users questions and get answers from them. However, when asking and answering questions, they need to consult prior research or examine asked about services. On the other hand, our method has the advantage that it can attach functions to Web sites. Nichols et al. [3] said that proactively contributing to usability activities of users is valuable. However, users could not sufficiently improve BADUIs only by functions in prior work because there are various BADUIs on the Web and many input forms are particularly confusing for users. Therefore, we propose an improvement function that allows users to add a conversion filter automatically to the input form. Also, our system can use several functions to improve many kinds of BADUIs.

3 WEPATCH

3.1 Proposed method

To solve BADUI's problems on the Web, we propose WePatch, a system that enables ordinary users to augment usability virtually by attaching improvement functions to a Web page. WePatch has five functions.

- **Auto-conversion filter**: This function prevents mistakes by automatically converting the inputted information into the correct format (see Fig. 1).
- **Display examples**: This function shows an input example on the input-form to help users input information efficiently (see Fig. 2).
- **Balloon description**: This function shows speech bubbles to explain unintelligible words such as technical terms (see Fig. 3).
- **Web advisory sticker**: This function presents clues (see Fig. 4).
- **Information correction**: This function corrects erroneous information on Web sites (see Fig. 5).

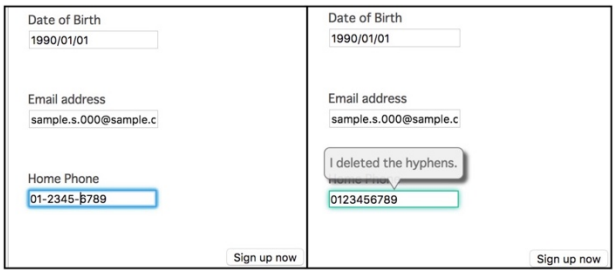

Figure 1: Auto-conversion filters for correcting input formats.

Figure 2: Presenting grayed-out input examples to other users.

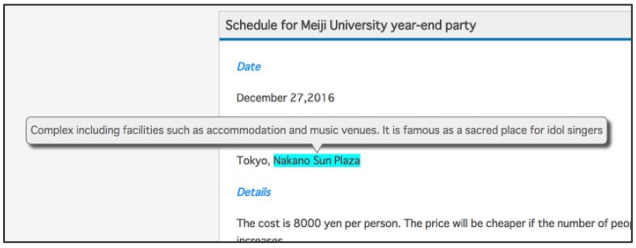

Figure 3: Referencing comments by other users.

Figure 4: Notifying users by affixing conspicuous stickers to Web pages.

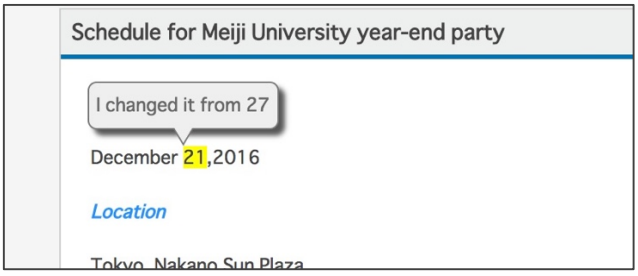

Figure 5: Correcting typos on Web pages.

Also, the users can share improved Web pages by using URLs and improving IDs. These allow users who have installed our system to access improved Web sites by themselves or by working with other users. Our method addresses the above problems (except the feedback problem described in Section 1) with these improvement functions. A page may receive these improvement functions from many users. If users can browse all the added annotations, a malicious user may use an erroneous operation to play users off against each other and make it hard for them to see Web pages. Therefore, our method links the Web pages and the function information by assigning a uniquely generated ID automatically to the original URL of a Web page. At the same time, our method adds an improved function and sends it to the database. For example, consider a case in which XXXX is generated as the ID of the target Web page such as "http://example.jp/test.html", so the improved page's URL is "http://example.jp/test.html#XXXX" with a hash value. A user can access the improved Web page only by accessing the page URL with this hash ID and by using our system. Then, the user will be able to access this improved Web page efficiently by bookmarking the URL with the hash ID or clicking the shared URL with the hash ID on the SNS (social networking service). Also, users can uniquely identify the improved Web page by the improved ID, so they can avoid security problems if the sender is clear (the Web site administrator, established engineers, parents, and so on).

3.2 Implementation

We implemented a prototype WePatch system with these functions as a Web browser extension for Google Chrome by using JavaScript. We also implemented a server-side system to store and share improved information as a Web API (application programming interface) by using PHP and MySQL. Our system reflects improvement functions among users who downloaded this extension in Google Chrome (see http://nkmr.io/wepatch/). JavaScript is used to perform processing such as insertion and replacement of elements in the DOM (document object model) of a Web site, and PHP uses asynchronous communication with Ajax to obtain and send values relevant to the function information (such as ID, URL, and DOM) to MySQL. The UI of the WePatch system is displayed in the bottom right part of the browser by an extended function form. When users click the button, abbreviated names of the eight functions are displayed. Users can use any of the improvement functions by selecting these buttons and can share improvement information by clicking the "Share" button. Users can also post BADUI information to share with others by clicking the "Tipster" button.

4 EVALUATION EXPERIMENT

To clarify the usefulness of WePatch, we evaluated the system in a test and compared the usability of Web sites before and after being improved by using WePatch.

First, we asked participants to find Web pages that they thought were complicated to use and to improve the pages with WePatch. The participants were ten undergraduate students of information science aged 20 to 22 (seven males and three females) who had had a lecture on BADUIs before the experiment. As a result, the participants added functions to 16 of the 23 gathered BADUI cases. (The 16 cases were eight for information provision, seven for input-form, and one for video viewing). Example deficiencies include a hard-to-find download button on a programming code sharing site was hard to find, a citation containing erroneous information, an input-form that did not provide sufficient information on what to input or how to input it, and an input-form where "Please input" above the unedited text box was an obstacle to input.

Participants used the Web advisory sticker most frequently. They used the auto-conversion filter and display examples often in the input-forms. Second, we gathered ten new participants under the same conditions as before and divided them into two groups (three males and two females belonged to one group; four males and one female belonged to the other) for evaluating BADUIs before and after improvement. We asked them to evaluate how easy the Web pages were for them to use. Fig. 6 shows the evaluations before and after improvement for each case. The y-axis shows the five Likert-scale [8] grades (-2: hard to use, -1: somewhat hard to use, 0: neither easy nor hard, 1: somewhat easy to use, 2: easy to use). The x-axis shows 16 BADUI cases (A to P). The graph shows that 10 of the 16 BADUI cases had scores lower than 0 before improvement and that 10 had scores of 0 or more after improvement. We found the scores improved by 1.08 points on average after improvement (-0.45 points before improvement; 0.20 points after improvement). Also, t-test results with $p<.05$ revealed significant differences between the scores before and after improvement. From these results, we conclude that using WePatch effectively improves BADUIs. On the other hand, scores decreased for cases B, G, and N. Also, some participants stated in the questionnaire that they felt the Web advisory sticker was an obstruction, citing reasons such as it was in a place that affected the operation, was in a place that was not particularly complicated to use, and gave unclear information.

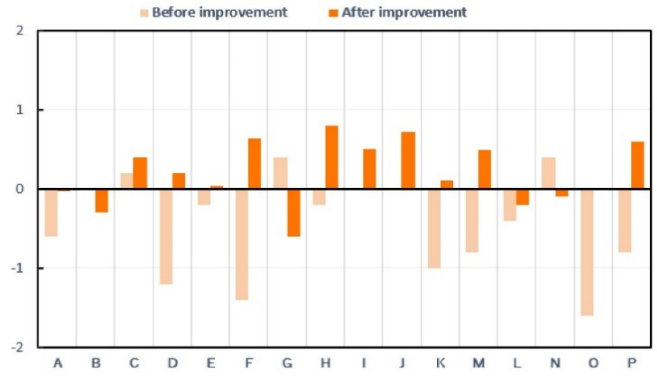

Figure 6: Comparison of evaluations before and after BADUI improvement.

5 DISCUSSION

The obtained experimental results show that the auto-conversion filter and Web advisory sticker tended to improve many BADUIs. The results show that these functions extend existing browser extensions to allow users to improve BADUI in the input forms. Accordingly, we believe these functions can resolve many BADUIs. One such problem is that many users waste time during registration because registration Web pages are often hard to use. Therefore, if someone uses our system to improve such Web pages and shares a URL with a hash ID among users, incorrect user operations will likely decrease.

To summarize the advantages of this system, ordinary users can use it to improve hard-to-use Web sites, and developers can use it to obtain clues to find hard-to-use pages and improve them in accordance with the functions the users attach. Also, Web site operators can recover their sites as users, making it possible to reduce expenses that occur when responding to improvement requests. Therefore, WePatch has advantages for not only users but also Web site operators, administrators, and developers. Also, using it to collect a massive number of BADUI improvement cases from the vast spaces on the Web makes it possible to accumulate knowledge about what kinds of Web sites are problematic and how they can be improved. In this study, we did not conduct an evaluation test on the Tipster function in our work. However, this function can help users to maintain their motivation for improving Web sites by enabling them to share the confusion and stress they have felt with others, sympathize with each other, and collect BADUI information for Web developers or researchers.

Our system cannot resolve BADUI issues such as the feedback and animation problems such as a lack of feedback at the time of error on the input form, an input form with an unknown lower limit of the number of characters, and moving animated elements that need to be clicked. Therefore, we plan to implement a function that can embed JavaScript and allow users to address many kinds of BADUI. Also, from results in Chapter 3, users should not try to improve BADUI issues allow us to some extent due to opposite effect. Therefore, we plan to conduct an in-depth experiment to clarify the property of the Web advisory sticker.

5 SUMMARY

We proposed and implemented WePatch, a system that enabled users to improve BADUIs (BAD User Interfaces) on the Web virtually and to share improved Web pages with others who have installed WePatch. We also concluded that WePatch is effective for improving BADUIs. In future work, we are planning to implement a function for identifying and evaluating contributors who have helped to improve Web pages and to judge their credibility. Also, we are also planning to collect information on improving procedures so that we can analyze and classify them to develop an automatic improvement method for BADUIs.

ACKNOWLEDGMENTS

This work was partially supported by JST ACCEL Grant Number JPMJAC1602, Japan and Meiji University Priority Research A.

REFERENCES

[1] Brenda Battleson, Austin Booth, and Jane Weintrop. 2001. Usability Testing of an Academic Library Web site: A Case Study. Comm. *ACM* 27, 3 (2001), 188-198. DOI: http://dx.doi.org/10.1145/1188913.1188915

[2] Dale Reed, and Sam John. 2003. Web annotator. Proceedings of the 34th SIGCSE technical symposium on Computer science education (SIGCSE '03), ACM 35, 1(2003), 386-390. DOI: http://dx.doi.org/10.1145/792548.612014

[3] David M. Nichols, Dana McKay, Michael B. Twidale. 2003. Participatory Usability: supporting proactive users. Proceedings of 4th ACM SIGCHI NZ Symposium on Computer-Human Interaction (CHINZ'03). 63-68. http://dl.acm.org/citation.cfm?doid=2331829.2331841

[4] Jason I. Hong, Jeffrey Heer, Sarah Waterson, and James A. Landay. 2001. WebQuilt: A proxy-based approach to remote web usability testing. ACM Transactions on Information Systems, ACM 19, 3 (Jul. 2001), 263-285. DOI: http://dx.doi.org/10.1145/502115.502118

[5] Jonathan W Palmer. 2002. Web site Usability, Design, and Performance Metrics. Information Systems Research. Comm. ACM 13, 2 (2002), 151-167.

[6] José Kahan, and Marja-Riitta Koivunen. 2001. Annotea: an open RDF infrastructure for shared Web annotations. Proceedings of the 10th international conference on World Wide Web (WWW'01). ACM 39, 5 (2001), 623-632. DOI: http://dx.doi.org/10.1145/371920.372166

[7] Ka-Ping Yee. 2002. CritLink: advanced hyperlinks enable public annotation on the Web. Computer supported cooperative work (CSCW'02).

[8] Rensis Likert. 1932. A Technique for the Measurement of Attitudes. Archives of Psychology. 140, (1932). 1-55.

[9] Satoshi N. 2015. Learning user interface from failure examples (in Japanese).

[10] Tenembaum Samuel S, and Ivanoff Ivan A. 2003. Enabling communication between users surfing the same web page.