



17th International Conference in Knowledge Based and Intelligent Information and Engineering Systems - KES2013

PastePost: A Web Interface for Editing Instructions with Captured Image

Motoki Miura^{a,*}, Shinya Yoshida^b

^a*Faculty of Basic Sciences, Kyushu Institute of Technology, Japan*

^b*Department of Integrated System and Engineering, School of Engineering, Kyushu Institute of Technology, Japan*

Abstract

In this paper, we introduce an effective Web-page construction method especially for instructions and tutorials of GUI based operations. We often obtain knowledge of specific software tools and its usages by Web documents such as blog entry and product sites. Also we sometimes need to create instructional web pages for teaching and distributing particular softwares. The instructional web pages contain several steps of operation with a text and a captured image. Organizing the instructional web page is time-consuming task because the editor need to prepare captured image files and upload them to the server. To relief the burdens, we propose a method of uploading clipboard images to the instructional web pages instantly. The method provides functions of (1) uploading captured images by a shortcut key, (2) instant image editor, and (3) organizing texts and images with wysiwyg manner. We performed an experiment regarding usability, and confirmed that the proposed method improves efficiency and satisfaction of the web organizing tasks.

© 2013 The Authors. Published by Elsevier B.V.

Selection and peer-review under responsibility of KES International.

Keywords: Page Builder; Web ; Step-by-step Instructions; Software Tutorials ; HTML5

1. Introduction

As an extent of casual information exchange including smart phones and tablets, various software tools appear and diffuse. The authors of the software tool often introduce their services and products in their blogs and web sites as well as official software stores. The blogs and web sites often contain many screenshot images to express actual usage of the software tool. Step-by-step instructions and tutorials also contain many images.

Most of blogs and CMSs which are used to build web pages on a web browser offer a file upload function. However, in order to build a web page which contains many screenshot images, the author should prepare many image files, upload them, and specify the place where and how these images shown on the web page. The page building task includes a large amount of image file management tasks, and it may limit the time to elaborate the document. Consequently, the quality of the document in total will be reduced.

*Corresponding author. Tel.: +81-93-884-3429 ; fax: +81-93-884-3429.

E-mail address: miuramo@mns.kyutech.ac.jp.

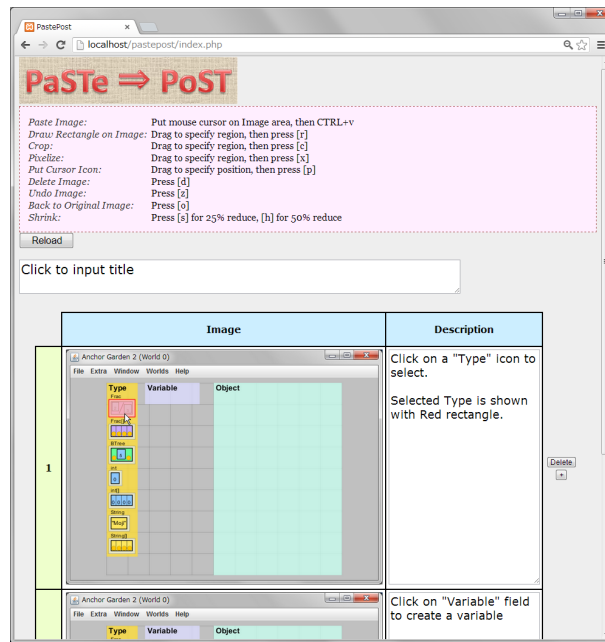


Fig. 1. PastePost interface. The page consists of table cells.

To relieve the burdens of managing many image files on a web page, we propose a method of uploading clipboard images to the instructional web pages instantly. The method consists of three functions: (1) uploading captured images by a shortcut key, (2) instant image editor, and (3) organizing texts and images with wysiwyg manner. The proposed integrated method simplifies the steps of file upload and management tasks.

2. Proposed Method

In order to reduce the file management tasks, the system should accept not only the saved image file but also the clipboard image directly. Because giving identical, consistent, and unique filename for multiple images is difficult. Also, the system should be implemented on the conventional web browsers. Since the specific client tool requires installing and setting of server connections, and it may reduce the opportunities of informal knowledge sharing.

Nowadays, most of the modern browser like Firefox and Google Chrome can support File/Drag and Drop/Clipboard APIs. Drag and drop file upload is familiar with GMail attachment. Pasting clipboard data including image becomes possible recently. Therefore, we decided to implement the three functions; uploading, editing and organizing images as HTML5 web applications.

We utilized techniques presented as jquery.paste_image_reader.js jQuery plugin ¹ to realize handling of pasted image. The pasted image is available in a callback function of pasteImageReader handler. When the image pasted, an ID of the focused table cell is detected by jQuery. Ajax (XMLHttpRequest) transmits the image (base64 encoded) and the ID which specifies the target place. Server side script decodes the base64 image, and stores as a PNG file. The stored images are soon be loaded by the browser, and displayed on the web page. The mouse event handlers are attached to the loaded image, and utilized for further editing such as cropping, pixelizing, resizing and so on. The cropping and resizing functions are important because the screenshot image tends to large for displaying on the page. The image manipulations are realized by PHP/GD functions on the server-side.

¹<http://strd6.com/2011/09/html5-javascript-pasting-image-data-in-chrome/>

3. Implementation

In this section, we demonstrate the system with screenshot images. Note that the system was running on an XAMPP local host LAMP server.

Figure 1 shows the editing interface of PastePost system. The page consists of a table with two columns. Pasted images are shown in the left column, and descriptions are on the right column.

The numbered row is inserted by clicking [+] buttons along with the table cell. Figure 2 displays three empty rows generated by user's request. When the user hovers the mouse cursor on a cell in left column, the color of the cell changes to gray (Figure 2 left). The color indicates the target cell. When the user presses CTRL+V at that state, the clipboard image is soon pasted and displayed on the target cell (Figure 2 right).

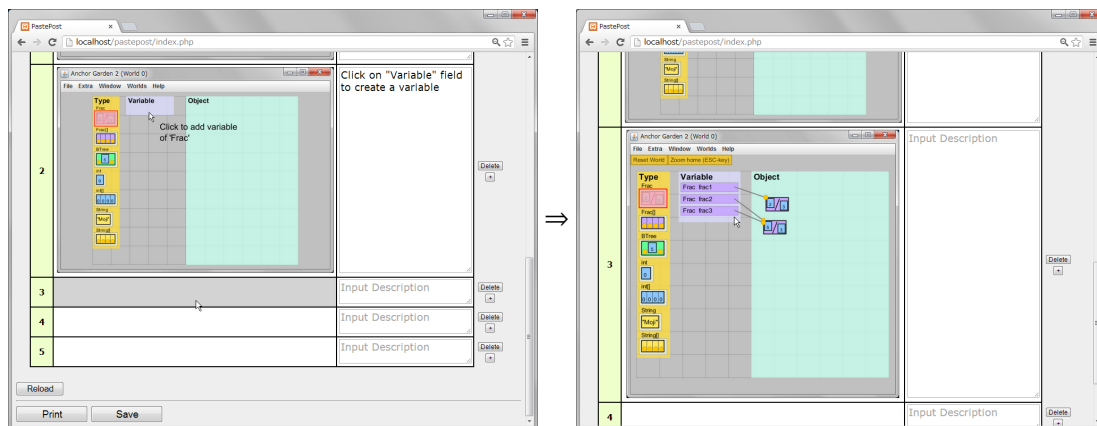


Fig. 2. The numbered row is inserted by clicking [+] buttons. Before the image pasting, put the mouse cursor on the target cell (left). When the user presses CTRL+V, the clipboard image is pasted on the cell (right).

The user can crop the pasted image at arbitrary rectangle. When the user drags the mouse, the guide rectangle appears (Figure 3 left). By pressing [C] key, only the region remains (Figure 3 right). Similarly, pressing [X] to pixelize the selected regions (see Figure 4), pressing [P] to put mouse cursor image, and pressing [R] to draw rectangle on the image. The pixelize function can be utilized to hide irrelevant portions of image. The putting cursor image and rectangle on the image are effective for highlighting the operating target such as buttons and choices.

If the user mistakes in image editing, PastePost system offers several undo functions. Pressing [Z] to undo last operation, and pressing [O] to revert to original uploaded image. The user can also delete uploaded image by [D] shortcut keys. Thanks to the undo functions, the user freely tries decorating on the images.

4. Related Works

We categorized the related works as the followings: uploading images, editing web pages on web browser, and automatic generation of tutorial.

4.1. Uploading Images

In the use of images on the Web page, the most important is the Web server to upload the image. As described in Section 1, in the process of browsing screen to upload images to choose from, there is a problem that it takes time and effort, however. Therefore, techniques have been developed with the aim of carrying out these processes more concisely.

Pasteboard developed by Joel Besada [1] is a quick and simple image sharing Web application. When the user drop or paste an image on the web page, the image is soon uploaded to the server, and the image and its shortened URL is displayed on the page. The image will be stored on the server for 90 days. Snaggy [2] also provides an image uploading service. Snaggy can edit the uploaded image by adding annotations by pen drawing and putting

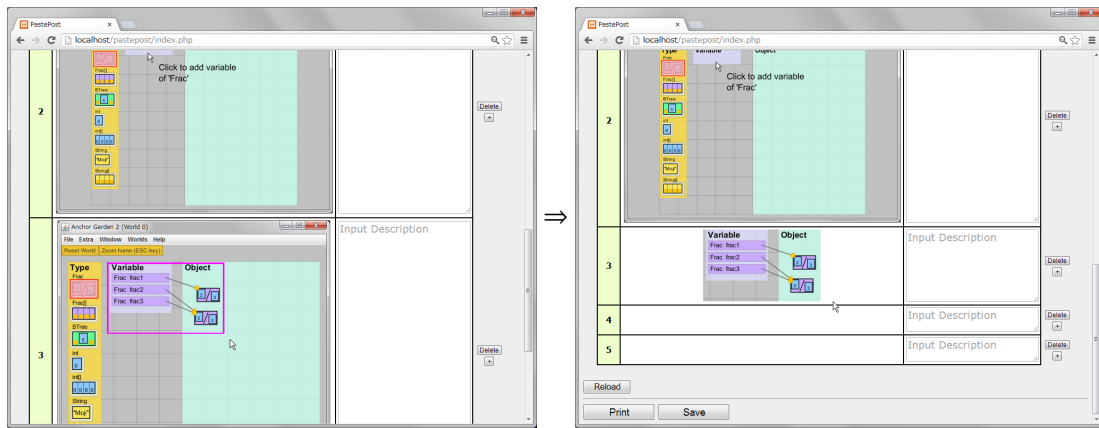


Fig. 3. Cropping by rectangle. The area is specified by the mouse dragging (left), then press [C] key to perform crop (right).

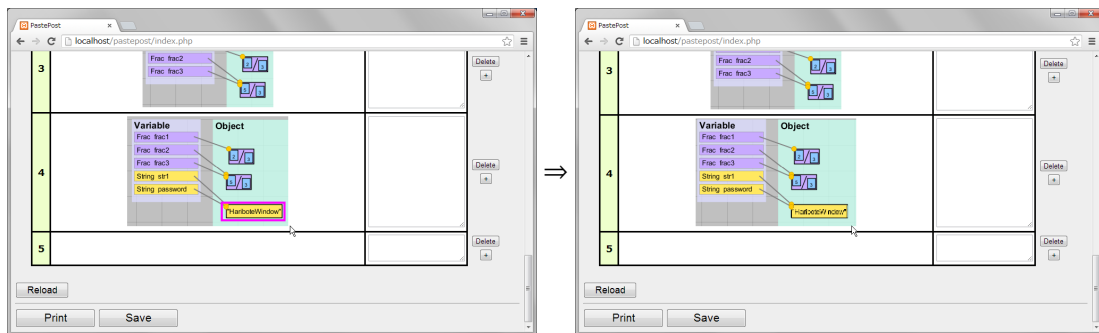


Fig. 4. Pixelizing by rectangle

text. Cropping is also supported on the web site. Both the Pasteboard and the Snaggy enables the user to upload clipboard image by a simple pasting operation. Gyazo [3] also provides a quick uploading of screenshots. By installing Gyazo client software on PC, the user can crop the desktop area by simple mouse selection. Thus the user can freely choose the area of the image even if it ranges multiple windows. These sites are designed to share a few images for referencing by other services like twitter and facebook, however PastePost can build a web page that contains large amount of images, without any extra services.

Zhang and Ren showed a solution to paste clipboard image directly on the page by extending CKEditor [4]. CKEditor is an open source WYSIWYG editor designed to simplify web content creation, and runs on a web browser. The concept of the work is similar, but we focus on the instructional page for simplicity, and provide further image editing functions.

4.2. Editing Web pages on Web browser

Generally, editing a web page is difficult for casual users. Wiki and Blog systems are solutions for editing and updating web sites for casual users. The user can update contents via a conventional web browser. Of course most of the Wiki and Blog system contain functions to upload image files, and specify the image displays inlined. However, there are many steps for publicizing the page with uploaded images.

NOTA [5] is a collaborative web platform that allows users to create, share and collaborate on presentations with virtually any form of online material. The users can instantly integrate text, video, maps, clip art, photos from the web, on your computer and from Flickr. The NOTA is suitable for freely arrangement of visual contents, but adding photo interface does not support paste operations.

Amaya Web editor [6] is a tool used to create and update documents directly on the web. Browsing features are seamlessly integrated with the editing, and remote access features in a uniform environment. The Amaya follows

the vision of the Web as a space for collaboration and not just a one-way publishing medium. Amaya supports editing of HTML, CSS, XML, MathML, SVG and more. The concept of Amaya is similar but it covers general web development. The PastePost covers limited target pages but it offers better experience by target specific, intuitive interfaces.

4.3. Automatic generation of tutorial

Grabler [7] generates succinct step-by-step visual tutorials of photo manipulation with screenshot images from author's demonstration. The system records the author's actions and the corresponding image processing operations in the application, and generates labels from the image changes. MixT [8] is a system that automatically generates step-by-step mixed media tutorials from user demonstrations. MixT separates screen-capture videos into steps by considering application logs and input events. The MixT tutorials are generated as standard Web documents. These automatic generating approach are quite effective for reducing tutorial author's burdens. However, the tool relatively depends on the target applications. PastePost covers general applications which provide GUI operations.

5. Experiment

We have conducted an experiment to verify the improvement of the usability of the system according to the PastePost. This chapter describes the procedure and results of the experiment and discussion.

5.1. Procedure

We asked 12 participants (all belonging to a student) to make instructional documents in two different methods. All participants were no experience to make the instructional document.

To evaluate the usability, we compared our system PastePost with Microsoft Office Word. The reason of choosing MS Word was (1) it is familiar with people for making documents, and (2) the user can paste an image by drag-and-drop of the image file. Although the MS Word can paste clipboard image directly, we asked the participants to use a capture software (we utilized CapWrite²) to prepare images. The software enables the user to capture, edit, and save screenshot images by simple operation.

We prepared two tasks regarding configuration of Windows OS for the experiment; changing wallpaper of desktop on Windows 7, and adding a timezone in Windows 7. We prepared standard procedure of these tasks by simple text instructions, and the instructions were provided to the participants as references. Also we prepared template documents that include a table and corresponding text instructions to reduce the burden of the participants. Thus the main tasks of the participants were to capture instructional images and paste them to the proper cells in the table.

To know the participants' expertise of the MS Word and PC, we asked the skills by 48 questionnaire items in advance. The questionnaire items were related to experience and knowledge of image handling on PC. The following were the examples of the questionnaire items: (1) Have you inserted a picture into a document? (2) Have you ever changed the size of the image pasted into document? (3) Have you ever upload images to blogs or SNS? (4) Have you ever use the PrintScreen key? (5) Do you know what the active window is? By considering the result, no significant differences were found in the participants' expertise. Therefore we treated the participants without special grouping.

The followings were tasks in the [Word] condition. (1) Read the provided instruction, and display the screen along with it. (2) Take a screenshot using CapWrite. (3) Edit the image by embedding a mouse cursor, resizing and cropping. (4) Save the image as a file. (5) Insert the image to the table by Drag&Drop. (6) Save the document as a PDF file. The tasks in the [PastePost] condition were: (1) Read the provided instruction, and display the screen along with it. (2) Capture the screen and copy the image to the clipboard. (3) Put the mouse cursor on a table cell, and press CTRL+V keys to paste. (4) Press "Save" when finished.

²<http://homepage1.nifty.com/chappy/lib/libcawr.htm>

5.2. Questionnaire

This experiment was designed to verify the usability of insertion and editing of images. We prepared questionnaire items in accordance with the definition of usability in ISO9241-11³. The usability definition consists of four categories: effectiveness, efficiency, satisfaction, and context of use. However, we eliminated items regarding to the effectiveness: “accuracy and completeness with which users achieve specified goals,” because both PastePost and MSWord have ability to achieve the specified goal. Thus we set the following items in the experiment.

- (After MSWord condition)
 - Did you satisfied with the completed document? (1-5)
 - Do you think the tool was easy to create the document? (1-5)
 - Do you think the task was performed comfortably? (1-5)
 - How much you want to use the tool for future similar work? (1-5)
 - Do you think how long does your work take.
 - Please tell if there is a feature you want to add. (free description)
- (After PastePost condition)
 - Did you satisfied with the completed document? (1-5)
 - Do you think the tool was easy to create the document? (1-5)
 - Do you think the task was performed comfortably? (1-5)
 - How much you want to use the tool for future similar work? (1-5)
 - Do you think how long does your work take.
 - Please tell if there is a feature you want to add. (free description)
- (After both conditions) Please tell how you select these tools. (free description)

We added the last question for gathering comments and reasons regarding how the tools were selected in some cases.

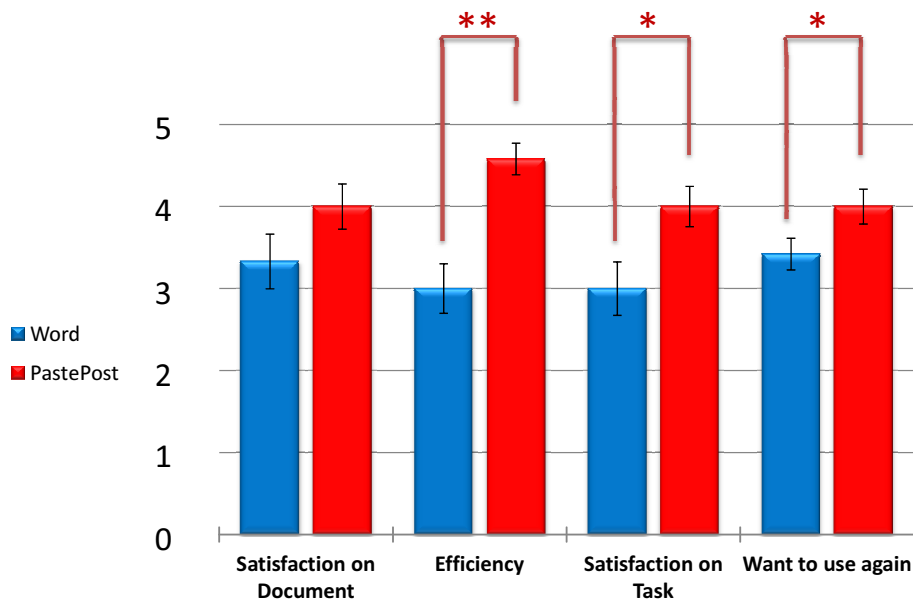


Fig. 5. Result of the questionnaire survey. ** denotes 1% significance, and * denotes 5% significance.

³<http://www.usability.gr.jp/whatis/definitions/>

5.3. Results and Discussion

Figure 5 shows average scores of the questionnaire by 12 participants. The four categories in the figure 5 correspond to the first four questionnaire items answered by (1-5) values. The error bar denotes standard errors.

Regarding the average scores, PastePost was higher than MSWord. We performed paired t-test for each category under 5% significance levels. Consequently, “Efficiency” ($t(11) = 4.42, p = .001$), “Satisfaction on Task” ($t(11) = 2.45, p = .032$), and “Want to use again” ($t(11) = 2.24, p = .046$) were significantly differed. No significant difference was observed in the “Satisfaction on Document.”

The reason of no significant difference on “satisfaction on the document” was that the task in this experiment was only uploading, and the participants could not fully customize their documents.

By comparing the measured actual time and the reported feeling time by the participant, PastePost was able to finish early than their feeling time. Also the average working time of PastePost condition was 52% of the MSWord condition. Considering these results, we can conclude that intuitiveness of PastePost method can surely reduce the burden of uploading and managing captured images.

From the participants’ free descriptions on how the tools will be selected, some participants asked that he/she utilize MSWord when he/she put emphasis on writing, and elaborate over time. On the other hand, PastePost will be used when the amount of image is large, and he/she wants to build a simple instructional page quickly. These comments will be influenced with their impression on these tools including their experience. The participants in this experiment were not a specialist of computer usage. Therefore they did not have the demand and experiment to publicize the instructional page enough. We will continue to assess the usability and collect the comments from specialists.

Regarding the features to be added to MSWord, putting typed text in an image and editing functions were required. MSWord already provides a function to put texts and figures on the document, but it is hard to maintain positioning between these items and images. Also the user of MSWord can crop images, change brightness, and so on. Similarly, most of requested features to PastePost were enrichment of image decoration and customizing of image shrink parameters. Many participants requested the pop-up menu of the image editing operations. We will improve the usability and the functions of PastePost for casual image-based knowledge exchange.

6. Conclusion

In this paper, we introduced a method of uploading clipboard images to the instructional web pages directly. The features of intuitive uploading and instant editing reduce the burdens of managing captured image files and specifying them on the target web pages. We also developed the PastePost system that implements the functions proposed by the method. Our experiment revealed that the advantages of PastePost by comparing with MSWord, in efficiency, satisfaction on task, and repetitive use.

Even though the functions are still limited, PastePost can offer better experiences to the web author without any software installation and configurations. The characteristic is suitable for casual posting and sharing of the instruction on the web. Moreover, comparing to the static PDF file generated by MSWord, the PastePost is beneficial because dynamic effects and expressions on the Web document, such as zooming and highlighting animations, are easily applicable. We will improve both the usability and the attractive representation of the document, and enhance the power and possibility of the world-wide web itself.

Acknowledgments

Our research is partly supported by a grant-in-aid for Scientific Research (23680078).

References

- [1] J. Besada, Pasteboard, <http://pasteboard.co/>.
- [2] T. Lee, Snaggy, <http://snag.gy/>.
- [3] T. Masui, N. Inc., Gyazo, <http://gyazo.com/>.
- [4] B. Zhang, M. Ren, Paste image from clipboard using javascript in ckeditor, *Advances in Computer Science and its Applications* 1 (4) (2012) 240–243.

- [5] I. Rakusai, Nota, <http://nota.jp/>.
- [6] INRIA, W3C, Amaya home page, <http://www.w3.org/Amaya/>.
- [7] F. Grabler, M. Agrawala, W. Li, M. Dontcheva, T. Igarashi, Generating photo manipulation tutorials by demonstration, *ACM Trans. Graph.* 28 (3) (2009) 66:1–66:9. doi:10.1145/1531326.1531372.
URL <http://doi.acm.org/10.1145/1531326.1531372>
- [8] P.-Y. Chi, S. Ahn, A. Ren, M. Dontcheva, W. Li, B. Hartmann, Mixt: automatic generation of step-by-step mixed media tutorials, in: *Proceedings of the 25th annual ACM symposium on User interface software and technology, UIST '12*, ACM, New York, NY, USA, 2012, pp. 93–102. doi:10.1145/2380116.2380130.
URL <http://doi.acm.org/10.1145/2380116.2380130>