System for Evaluating Usability and User Experience by Analyzing Repeated Patterns

Young Bin Kim[1], Shin Jin Kang[2], and Chang Hun Kim[3]

[1] Interdisciplinary Program in Visual Information Processing, Korea University, Korea
zulzin@korea.ac.kr
[2] School of Games, Hongik University, Korea
directx@korea.ac.kr
[3] Department of Computer and Radio Communications Engineering, Korea University, Korea
chkim@korea.ac.kr

Abstract. In this paper, a new system for evaluating interface usability through the analysis of repeated patterns is proposed. The system can be a valuable tool for verifying interfaces and in evaluating their usability by users, both of which are necessary stages in the development and operation of software. This paper concentrates on the repeated patterns that occur when users use an interface. Extracting these repeated patterns and analyzing them could enhance the development and usability of interfaces. Through experiments that applied the proposed system to several kinds of software, it was confirmed that problems with interfaces can be understood, and usability can be improved without requiring complicated analyses of user logs.

Keywords: Usability Methods and Tools, Analyzing Repeated Patterns.

1 Introduction

Evaluating usability is regarded as an essential element in the development of software. Such usability evaluation enables the software to be used more easily and efficiently, with users achieving their objectives and being satisfied. Moreover, by evaluating usability during development, any problems with the usability or functionality of the software can potentially be solved at an early stage.

Recently, many studies on the evaluation of usability have been undertaken. These studies mainly involved confirming the completeness and practicality of web designs or evaluating game interface designs [1, 2, 7]. The approach was usually to analyze the data obtained from surveys or log files, which recorded the activity of users, or to use technology such as user eye tracking [3-6]. Studies based on surveys have the advantage of easy and convenient measurement, but there is a possible problem in that subjects can distort the results, consciously or unconsciously. In contrast, observation of the reactions of a living body using eye tracking is advantageous for obtaining objective data, but the requirement to wear a special apparatus may cause unnatural behavior in subjects.

A. Marcus (Ed.): DUXU/HCII 2013, Part I, LNCS 8012, pp. 322–329, 2013.

In this paper, we propose a new system that automatically analyzes users' input log data to measure usability. The suggested system offers the advantage of collecting data naturally from many users and analyzing them automatically. This system mainly studies the aspects of the user interface that users focus on and the repetitive patterns that occur when the system is used. If a repetitive pattern exceeds a certain length, it indicates a poor user experience with the interface, or it warns designers that the specification of the interface should be improved. Our system analyzes sequences of user input data, visualizes them, generates tables of character-string patterns, and identifies repetitive patterns. We are able to evaluate the usability and the completeness of the application interface automatically via this system.

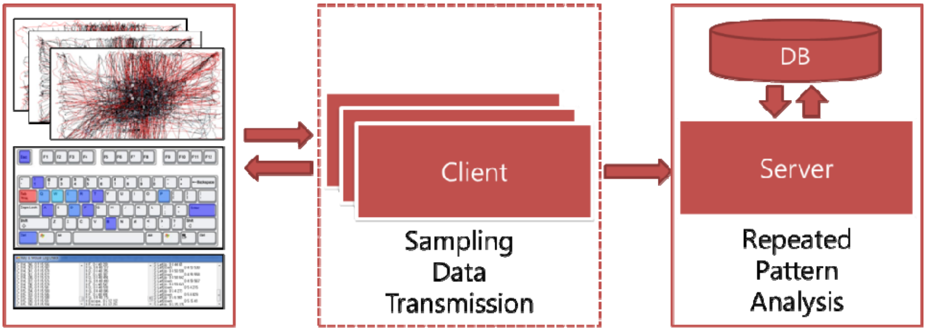

Fig. 1. System overview. Collect user input data from mouse and keyboard (Left). Transmit the received data from client to server (Middle). Conduct the usability evaluation by analyzing patterns repeated by the user (Right).

2 System

2.1 System Overview

The proposed system is based on a client–server architecture. It deals with sequences sent to the server during fixed periods while the users are generating mouse and keyboard input data (see Figure 1). The system has four main parts, a sampling part to collect the data from users, an identification part to find repeated patterns in the collected data, a visualization part that presents various pieces of information visually, and an analysis part to analyze the repeated patterns found.

First, in the sampling part, we collect data from the users' mouse and keyboard input over a fixed period of time and save it for repetitive-pattern identification analysis. For the mouse input, we classify the direction of movement as one of sixteen directions. For the keyboard input, we save the users' input as a continual sequence. In addition, we classify repetitive patterns as repetitive keyboard input sequences, repetitive mouse movements, or repetitive sequences based on a mixture of keyboard and mouse data (see Section 2.2).

Second, to search for repeated patterns in the keyboard and mouse input sequences, we use a method that updates a simple form of string-pattern table. We choose a minimum pattern length for pattern verification, and we are able to identify repetitive patterns that are above that minimum pattern length. In addition, by storing the coordinate values for these repeated patterns, they can then be visualized, enabling repeated patterns to be confirmed visually (see Section 2.3).

Third, the visualization part is designed to identify those aspects of the interface that users mainly focus on when they use the application. The input values from the mouse and keyboard are both visualized. For the mouse, its path and speed of movement are presented visually. This visualization enables understanding of what aspects of the system users tend to concentrate on. In addition, we found a way to identify the interface's utilization by marking the keyboard inputs following mouse movements. The duration of input standby times was also visualized for analysis (see Section 2.4).

Finally, the analysis part is where the repeated pattern table is used to identify patterns that may have potential problems (see Section 2.5).

2.2 Sampling

There is sampling of the input from the keyboards and mice of users. The sampling of the keyboard input is implemented in terms of sequential input values. For typical keyboard input sequences, additional (modifying) keys can be used, and the times when these keys are in operation are separately sampled. This gives information about the weight to give to the keys used, and enables identification of periods when many keys are used simultaneously.

As mentioned, the mouse input is classified in terms of sixteen directions, and this direction of movement is sampled. Here, we use directions that are pre-learned based on the Hidden Markov Model method, which minimizes the error values in mouse directivity. The keyboard input is sampled at the same rate for all input conditions, but the mouse input sampling is variable. An appropriate sampling rate will depend on the environment of the software to be evaluated. In this paper, software involving much mouse manipulation has a sampling period of 30 ms; for other software, sampling occurs every 50 ms.

Incorrect repeated patterns can be generated in mouse usage areas moving from work space to interface (ex: movement to one direction only). To solve such problems, the part where simple movement occurs mainly after data gathering has to be identified. In this paper, a region including more parts moving faster than others was set as the movement part.

In addition, the input standby time and the number of keyboard and mouse operations per minute are sampled and made available for use if necessary.

2.3 Repeated Pattern Table

By analyzing the sampled data, repeated patterns are found. Because the amount of sampled data can become large over time, resulting from an accumulation of users, it can be difficult to extract useful data. Recently, many studies in the data-mining field have involved this concept of "big data".

Sequential pattern-matching techniques are used to search for mutual correlation among items occurring in sequences by adding the concept of time to the association rules. GSP and SPADE are examples of these algorithms [8]. Such algorithms aim to find frequent sequences in a database.

However, it is difficult to use algorithms such as these for the extraction of sequential patterns in a normal software-development environment. In such an environment, identical and restricted keyboard input is used repeatedly and there is much repetitive mouse movement. Repetitive and tautological key values are often taken as sequences, which is a drawback to using the existing algorithms.

In this paper, a method that creates and updates a repeated pattern table is used. The method is advantageous because it can extract frequently occurring repeated patterns efficiently, and at low cost. An algorithm for creating a repeated pattern table is as follows:

```
1. Initialize the pattern table to contain all strings of
length one.
2. Take initial pattern with length 2 or more.
3. Add pattern to dictionary.
4-1. If pattern already exists, add next sequence to
pattern.
4-2. Else initialize a new pattern from second sequence
of previous pattern with length 2 or more.
5. Iterate 3 to 4 until the input sequence is exhausted.
```

If the client sends the sampled data to the server regularly, this algorithm enables the repeated pattern tables to be continuously updated and managed. Also, by managing the pattern tables, using a basic data compression algorithm, the time and memory space were not greatly affected.

2.4 System Visualization

The visualization part of the system allows you to identify those parts of the application that users tend to concentrate on and any problems in using the interface. This was inspired by IOGraph software [9], which was visualized using the movement and standby time for users' mice. The movement of the mouse of users is visualized in terms of its weighted speed. Weighting the speed is helpful in understanding where the users are concentrating their attention. In the system described in this paper, a speed of at least 30 pixels per second is weighted. If a heat map is used for all of the mouse movements, there is the problem of separating movements related to the work of the users from movements caused by an object of interest. To identify the degree of concentration on the user interface, a color map showing mouse movement speed is available (see Figure 2). An additional heat map was created and visualized for keyboard input.

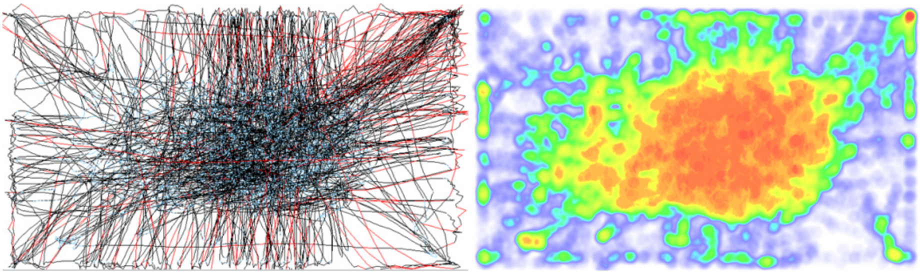

Fig. 2. The color map of mouse movement speed (Left) and the mouse movement's heatmap (Right)

2.5 Analyzing Repeated Patterns

In creating a repeated pattern table, it is important to understand what patterns have potential problems. In the present system, repeated patterns are analyzed in two stages. First, in the server stage, the sequences of all users are put together to identify repeated patterns. Second, in the client stage, the repeated patterns of particular users are analyzed. The repeated patterns for all users can indicate a problem with the interface on many occasions.

Through the system visualization, we can see that there are places where users are concentrating on their work using the mouse and places where there are simple movements.

3 Experiment

3.1 Experiment Design

A major experiment was carried out to evaluate the usability of a program currently being developed. To evaluate the completeness and usability of the interface of the software being developed, its usability was evaluated using the proposed system. The pieces of software used in the experiment are editing tools for general-purpose multimedia, used mainly for image and audio editing. This software supports numerous functions and most work is carried out using the movement and clicking of a mouse and the keys on a keyboard. Identifying problems with its interface could allow us to enhance its usability. The evaluation of this program was divided into three experimental periods, with the software being modified after each period. We confirmed that an improvement in actual usability was possible after the survey in each experiment. In addition, the usability of a commercial image-editing program was evaluated using our system.

3.2 Experiment Process

Experiment was carried out two times as a whole. In the first stage, the interface was improved using a repeating pattern table; in the second stage, usability evaluation by comparison between the interfaces before and after improvement was carried out. 4

participants were present at each experiment. They were all postgraduate students, recruited on campus, and were familiar with image processing, audio processing, and mouse and keyboard interfaces. They were given proper compensation for the participation.

The first experiment involving software under development lasted nine days. The software, with its prototype interface, was freely used for three days. From the results of the pattern table generated, the interface to the software was modified, and then the experiment was repeated for another three days. The basis of interface change was that of removing repetitive patterns. Shortcut keys were created for mainly appearing patterns or the relevant actions were placed on the top menu bar. The interface was again modified, and the process repeated for a third time.

In the second stage, the participants were instructed to perform specific tasks using the modified interface and asked the following (all measured in the 7 Likert scale) [10] :

- Learnability – Was there any difficulty in learning the interface?
- Ease of Use – How convenient was it to use the interface?
- Efficiency – How efficient was it to perform the specified task using the interface?
- Satisfaction – Was the interface satisfactory as a whole?

4 Result

4.1 Experiment Result

Through the experiment of the first stage, the locations of bottlenecks for the users of the software were identified, and the interface was modified appropriately. The figure 3 illustrates the survey result through the two test stages. We could verify that the users' satisfaction improved gradually with use of the modified interface.

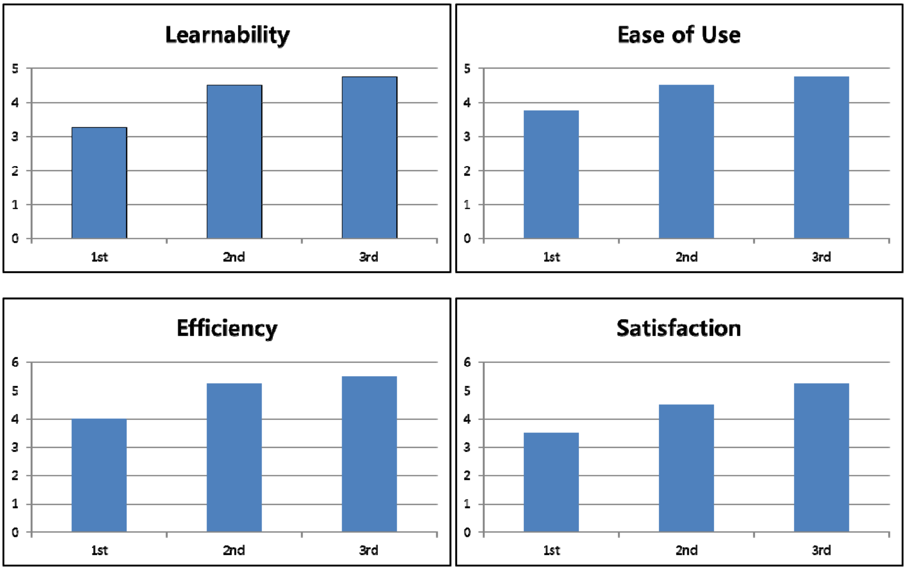

Fig. 3. Experiment results

Additionally, when particular users produced many repeated patterns or movements of the mouse, we could suggest ways for users to improve their usability.

4.2 Limitations

Because the experiment was undertaken with software under development, sufficient participants for a suitable evaluation of usability could not be recruited [11]. However, because of its server-client structure, it may be possible to obtain data for numerous users in the future. In addition, we plan to apply this system on software of various areas.

5 Conclusion

In this paper, we have analyzed input log data from various users and proposed a system that measures usability. The main advantages of this system are that it reduces the time required to investigate all log data and conducts usability evaluations effectively. Moreover, the proposed system is suited to the analysis of large datasets, and it will help improve user-interface designs. We believe this system will improve the evaluation of software usability in a variety of fields.

Acknowledgment. This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) and was funded by the Ministry of Education, Science, and Technology (NRF-2012R1A1A1012895).

References

1. Arroyo, E., Selker, T., Wei, W.: Usability tool for analysis of web designs using mouse tracks. In: CHI 2006 Extended Abstracts on Human Factors in Computing Systems, pp. 484–489. ACM (2006)
2. Atterer, R., Wnuk, M., Schmidt, A.: Knowing the user's every move: user activity tracking for website usability evaluation and implicit interaction. In: Proceedings of the 15th International Conference on World Wide Web, pp. 203–212. ACM (2006)
3. Mueller, F., Lockerd, A.: Cheese: tracking mouse movement activity on websites, a tool for user modeling. In: CHI 2001 Extended Abstracts on Human Factors in Computing Systems, pp. 279–280. ACM (2001)
4. Isbister, K., Schaffer, N.: Game usability: Advancing the player experience. CRC PressI Llc (2008)
5. Renshaw, T., Stevens, R., Denton, P.D.: Towards understanding engagement in games: An eye-tracking study. On the Horizon 17(4), 408–420 (2009)
6. Poole, A., Ball, L.J.: Eye tracking in HCI and usability research. In: Ghaoui, C. (ed.) Encyclopedia of Human-Computer Interaction (2006)

7. Kjeldskov, J., Stage, J.: New techniques for usability evaluation of mobile systems. International Journal of Human-Computer Studies 60(5), 599–620 (2004)
8. Agrawal, R., Srikant, R.: Mining sequential patterns. In: Proceedings of the Eleventh International Conference on Data Engineering, pp. 3–14. IEEE (1995)
9. IOGraphica, http://iographica.com/
10. Nielsen, J., Hackos, J.T.: Usability engineering, vol. 125184069. Academic press, San Diego (1993)
11. Hwang, W., Salvendy, G.: Number of people required for usability evaluation: the 10±2 rule. Communications of the ACM 53(5), 130–133 (2010)