

Two-level user interface evolution by sharing usage logs

Shuuji Morisaki¹ Akito Monden¹ Ken-ichi Matsumoto¹ Katsuro Inoue^{1,2} Koji Torii¹

¹ Graduate School of Information Science, Nara Institute of Science and Technology

² Graduate School of Engineering and Science, Osaka University

8916-5 Takayama, Ikoma, Nara 630-0101 Japan

{shuuji-m, akito-m, matumoto }@is.aist-nara.ac.jp,

inoue@ics.es.osaka-u.ac.jp, torii@is.aist-nara.ac.jp

1. Introduction

In this paper we propose a new framework for UI (user interface) evolution of software. UI evolution is a concept that UI should be continuously improved in response to the environment in which software has been used. Most of commercial software, such as a word processor, is supposed to be used for a wide range purpose. Yet, most software users actually use software for various purposes. For example, one may mainly use a word processor for writing a letter while others may use it for writing numerical formulas in a research paper. We claim that such difference in the usage of software should be considered in the UI evolution.

However, traditional UI improvement is based on the usability testing that uses only few subjects[4]. Because of variety in actual software usage, such traditional style of improvement may not always satisfy the software user who actually uses software.

The goal of this research is to evolve user interface that takes account of the environment where software is used and how software is used. Our approach is to evolve user interface by collecting each user's operation log through Internet. In this paper we propose two levels of UI evolution. First level evolution is to customize or to adapt a user interface by using local operation log. Second level evolution is to customize or to adapt a user interface by using global operation logs gathered through network.

Even though user logs are useful for UI evolution in response to the environment, we claim that using only local logs is not enough for UI evolution. It is because, a user may not always become aware of a useful function or operation. However, even in such case, there may be other user who is aware of useful functions and operations. In this situation, both local logs and global logs should be used for UI evolution.

2. Usage logs

Usage logs are series of user's operation to software, such as clicks on buttons and menu items. Logging user's operations provides his/her own usage trends such as:

- frequency in use of operations, and
- repetition patterns of operations.

Frequency in use of operations will be used for evolving a user interface of a single user. For example, split menus [5] organize frequency of menu selections. In split menus, a menu item is copied to the top of the menu if the menu item is often selected.

Repetition patterns of operations can also evolve one's own user interface. If a certain pattern of operation was repeated, series of the operation will be stored as a *macro*. After a macro was made, the macros will be invoked by a single click of a menu item.

3. Two levels of evolution

Evolution of user interface is classified into following two levels of evaluation.

- local evolution, and
- global evolution

As we mentioned in the previous section, local evolution is performed by using a single user's usage log. On the other hand, Global evolution is performed by sharing multiple users' usage logs.

Figure 1 shows a concept of macro sharing. We assume that users' computers are connected to the server through network. If a user has macros created by local evolution, then macros will be sent to the server for getting his/her consent. The server stores gathered macros. Each user can access stored macros.

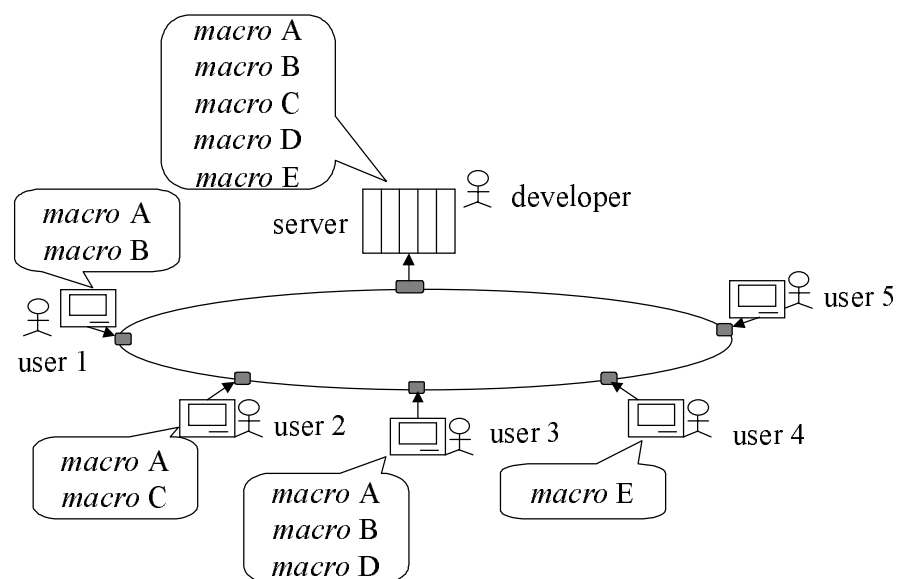


Figure 1. Sharing usage logs

Figure 2 shows global evolution. In this example, *macro A* consists of an operational sequence, button X, button Y, and menu item Z; and, *macro A* is stored in the server. If a user 5 clicks button X and button Y, he/she is noticed that others used *macro A*. He can choose either making *macro A* as his own macro and execute *macro A*, or ignore noticed *macro A* and continue his task.

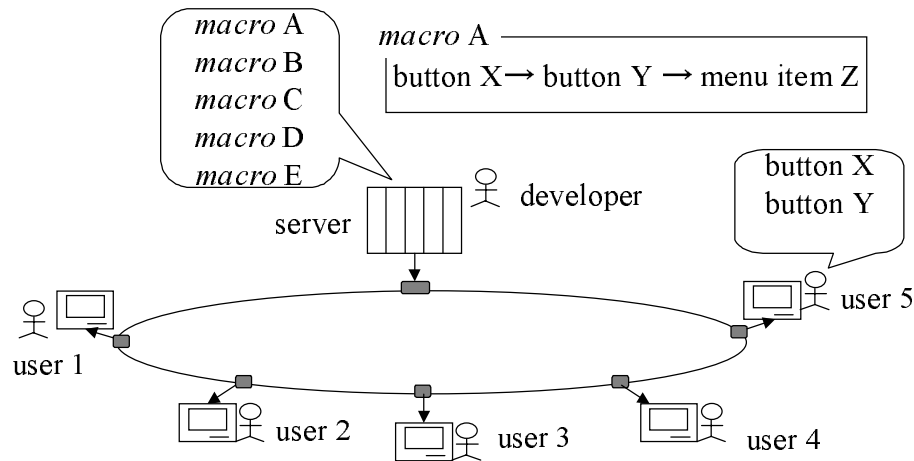


Figure 2. Global evolution

4. Discussion

Many researchers have been studied on local user interface evolution as adaptive or predictive user interface [1][2][3]. On the other hand, global evolution supports users

- to notice useful functions that a user is not aware, and
- to find fewer steps for achieving a task.

However, macros potentially contain noises such as useless or meaningless operations. If context of usage of macros is added, noises can be reduced.

When the new version of software is designed for local and global evolutions, this allow software developers:

- find the unique usage and to make it a new function of the software from analysis of someone's local evolution, and
- not to remove popular user interface or functions from most user's global evolution.

5. Conclusion

We have proposed two levels of user interface evolution using usage logs. Local evolution customizes single user's user interface. Global evolution enables users to share knowledge of the usage of software and to enhance the usage of the software. Local and global evolutions can support not only users but also the software developers.

REFERENCES

- [1] J. J. Darragh, I. H. Witten and M. L. James, "The reactive keyboard: A predictive typing aid," IEEE Computer, Vol.23, No.11, pp. 41-49, 1990.
- [2] S. Greenberg and I. H. Witten, "Adaptive personalized interfaces – A question of viability," Behaviour and Information Technology, Vol.4, No.1, pp.31-35, 1984.
- [3] T. Masui and K. Nakayama, "Repeat and predict – two keys to efficient text editing," Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '94), pp.118-123, Apr. 1994.
- [4] J. Nielsen, "Usability Engineering," Academic Press, 1993.
- [5] A. Sears and B. Shneiderman "Split menus: Effectively using selection frequency to organize menus," ACM Transactions on Computer-Human Interaction, Vol. 1, No. 1, pp. 27-51, 1994.