Meeting Scheduling System using Unpleasant Notification

Tomoyuki Mishima
Graduate School of Engineering, Tottori University
4-101 Koyama-Minami, Tottori 680-8550, Japan
Email: s082054@ike.tottori-u.ac.jp

Kenichi Takahashi
Graduate School of Engineering, Tottori University
4-101 Koyama-Minami, Tottori 680-8550, Japan
Email: takahashi@ike.tottori-u.ac.jp

Takao Kawamura
Graduate School of Engineering, Tottori University
4-101 Koyama-Minami, Tottori 680-8550, Japan
Email: kawamura@ike.tottori-u.ac.jp

Kazunori Sugahara
Graduate School of Engineering, Tottori University
4-101 Koyama-Minami, Tottori 680-8550, Japan
Email: sugahara@ike.tottori-u.ac.jp

***Abstract*—We often fail to arrange the scheduling of a meeting because someone does not input his/her schedule. Therefore, we hope a system which enables users to compel the input of his/her schedule. In this paper, we propose a system to compel users to input their schedules by using unpleasant notification. The system, first, uses a general notification, such as an e-mail, to ask to input their schedules, however, the system gradually uses unpleasant notification. Thus, the users are not unpleasant if they finished to input soon. Otherwise, they become gradually unpleasant because of unpleasant notifications. By using our system repeatedly, users will tend to input their schedules rapidly to avoid to be unpleasant.**

I. Introduction

Groupwares and some Internet services provide schedule arrangement services. For example, Chosuke [1] send an e-mail including the candidate days of a meeting to participants. Then, their participants input their schedules. We, however, often fail to arrange a meeting schedule because some participants do not input their schedule. Why do participants not input their schedule? They are some possible causes: they does not notice the e-mail; they forget to input; they feel troublesome to input, and so on. We have to eliminate such a causes to promote the input.

We can alleviate them by devising a user interface. For example, Google Chrome and Internet Explorer automatically input a password, a user Id and address and so on. This alleviates the troublesome of users. It is also effective to do the subject of an e-mail as [reply deadline:], and to improve visualization of the user interface. These can alleviate causes not to input, their effectiveness are limited because they does not compel users to input.

More effective method is to give an incentive to input. For example, a system gives much money to users who finished an input soon; few money to late input users; no money to users who does not input. Since sooner input conducts much incentive, we can expect more rapid input. It, however, has a big problem, who prepare the incentive. In this example, we have to prepare much cash. Thus, it is not realistic to use incentive. Furthermore, since the incentive must be gradually cut down, it is difficult to take additional countermeasure against users who does not input.

Therefore, we propose a system using unpleasant notification. The system, first, uses a general notification to ask to input their schedules, however, the system gradually uses unpleasant notification. For example, at first, the system sends only an e-mail. When the deadline of the input is approaching, the system reports user's lazy action to the boss of the user who has not inputted yet. Further, the list of users is written in the bulletin board as the users who have not input yet. The users are not unpleasant if they finished to input their schedules soon; however, theirs become gradually unpleasant. By using our system repeatedly, users will tend to input their schedule rapidly to avoid unpleasant situation. Our system does not give benefits to users, but gives unpleasantness. The human being reacts to a loss sensitively more than a gain[2]. Thus, the system using a loss such as unpleasantness will work more effective than a system using a incentive.

The remainder of this paper is structured as follows. Section II analyzes causes why users do not input their schedules. Section III presents our meeting scheduling system using unpleasant notifications. Section IV shows a preliminary experimental result. We show some related works in Section V and concludes the paper at Section VI.

II. Analysis of the Steps for Input

There are four steps to be passed for that a user finishes an input. Figure 1 shows their four steps and obstruction factors in each step.

Awareness: At first, it is necessary to inform that an input is required to a user. An e-mail is most popular way to inform the notification. It, however, requires a user to run an e-mail client, to operate it, and to read the notification. The user never notices it without their operations. Furthermore, the notification e-mail may be filtered out in a spam mail box. Therefore, we have to equip some notification ways.

Recognition: Even if a user receives a notification, the user never input his/her schedule if the user does not recognize

978-1-4799-2845-3/13/$31.00 ©2013 IEEE

that users input their schedule on a web site. Then, the work of a user is temporarily suspended, however, the user sees its web page certainly. Therefore, the user will notice the input is required.

Functional restriction: The system restricts any functions except seeing the web page for an input through a web browser. Then, a user cannot do any work except inputting his/her schedule. This is most effective to compel users to input, but has a big side effect.

We combine these notification ways to compel users to input their schedules.

C. Effectiveness of Each Notifications in Each Steps

There are four steps, awareness, recognition, its keeping, and execution, to be passed for finishing an input.

In the awareness step, it is important to highlight a notification. E-mail notification requires users to operate an e-mail client. Therefore, the users cannot notice the notification without its operation. Since pop-up notification is displayed on the screen automatically, the users is easy to notice it. Furthermore, a big pop-up is more noticeable than a small pop-up. Of course, these repeat the notification is more noticeable. Other two notifications can be awared certainly.

In the recognition step, it is important to read a notification carefully. Since e-mail and pop-up notification display only the subject of the notification, they are not so much effective. Switching focus on a web browser shows the content of the notification, however, it is ineffective if the user closes the browser immediately. Functional limitation can compel the users to read the notification because the users cannot do anything except finishing the input.

Main concern in the keeping step is to forget the input. It is effective to repeat a notification.

The final step is to input his/her schedule. In this step, it is effective that a user believes to be more unpleasant if he/she does not finish the input. Therefore, the system used notification way from little unpleasant way to much unpleasant way, first, uses one e-mail and/or pop-up notification, then, repeats them. After that, swithcin the focus on a web brower, finally, functional restiction.

TABLE I
EFFECTIVE NOTIFICATION IN EACH STEP

Awareness	E-mail $<$ Pop-up $<$ Switch the focus $<$ Functional restriction
Recognition	E-mail $\fallingdotseq$ Pop-up $<$ Switch the focus $<$ Functional restriction
Keep Recognition	E-mail $\fallingdotseq$ Pop-up $\fallingdotseq$ Switch the focus $<$ Functional restriction
Execution	E-mail $\fallingdotseq$ Pop-up $\fallingdotseq$ Switch the focus $<$ Functional restriction

D. Flow of the proposed system

E-mail notification and pop-up notification are popular mechanisms and used frequently. Therefore, users cannot feel unpleasant to them. In contrast, since other two notifications suspend users' work temporarily, the users feel unpleasant. It is undesireable to give unpleasant to the majority of the users. Therefore, we should gradually change notification way from little unpleasant way to much unpleasant way. Thus, the system, first, sends an e-mail for requesting the input of schedule to users in a meeting participant list. If a user input his/her schedule, the system removes him/her from the meeting participant list. The system repeats same action a few times at regular intervals. Next, the system performs pop-up notification to users who have not input yet. Since the pop-up notification does not required running a specific application, (the subject of) the notification is displayed directly on users' screen. The system repeats this action while changing a small pop-up to a large pop-up.

After that, switches the focus on the web browser and shows the web page for the input of a schedule. Then, the work of a user is temporarily suspended, however, he/she will notice certainly that an input is required. This action is also repeated a few times.

When the deadline of the input is approaching, the system restricts any functions except seeing the web page for an input through a web browser. Since a user cannot do any work except inputting his/her schedule, the user has to input his/her schedule. Finally, all schedules of participants must be gathered.

As shown above, users who input their schedules soon receive only one e-mail notification. Therefore, they never feel any unpleasant. However, lazy users who do not input increase the degree of unpleasant gradually. As a result, a user tries to avoid being unpleasant by finishing the input as soon as possible.

IV. EXPERIMENT

We have implemented the system as a Java-based system. E-mail notification has been implemented by using JavaMail. Pop-up notification has to use OS-specific API because it depends on each OS. Now, we have implemented the pop-up on Windows and Mac OS. Running a web browser has been implemented by using java.awt.Desktop. Functional restriction has been implemented by Selenium WebDriver2. Figure 3 illustrates the flowchart of our proposed system.

We compare time to finish the input of each users between our proposed system and using only e-mail notification. In this experiment, we set up the deadline of the input is on the noon four days after. First notification is sent out at 16:10 of the first day. In our proposed system, the system uses

- Day 1: E-mail notification,
- Day 2: Pop-up notification,
- Day 3: Switch the focus on a web browser, and
- Day 4: Functional restriction.

Nine users use our proposed system, and 24 users use only e-mail notification. The result of this experiment is shown in Figure 4

In our proposed system, all of users finished to input within 24 hours. On the other side, in only e-mail notification, only

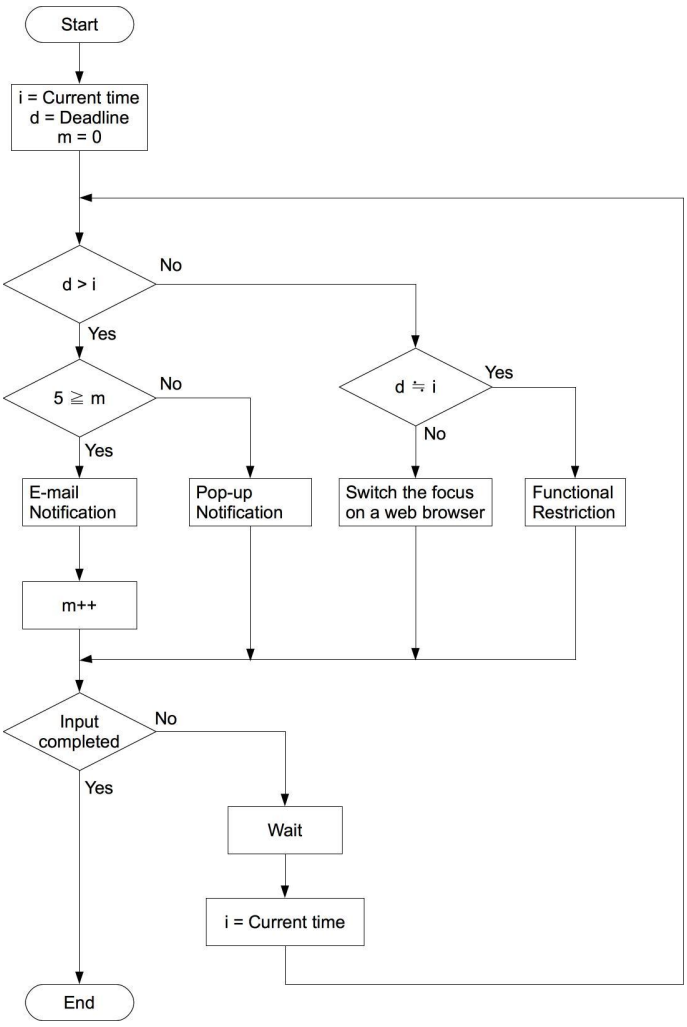

Fig. 3. Flowchart

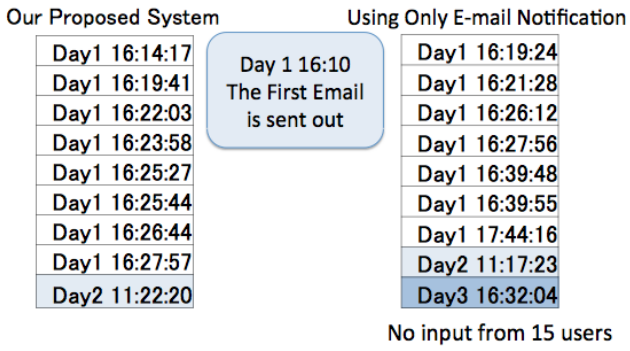

Fig. 4. Time of the input

nine users out of 24 users finished to input until the deadline, and fifteen users have not inputted.

V. Related work

[3] introduces a method based on behavioral economics to raise self-motivation for achieving some goal. In this method, there are a subscriber and a referee. The subscriber wants to achieve a goal, but is difficult by his/her motication. The referee keeps watch whether the goal is achieved by the subscriber or not. First, the subscriber offers a contract detail (e.g. losing 1 kg weight in a month) and a punishment (e.g. paying $1000) of when the contract is not achieved. Since the subscriber receives the punishment when he/she does not achieve the goal, he/she can raise his/her motivation is provided on Stick.com [4]. Such an actual service.

In [5] and [6], a user interface causing discomfort has been proposed. These analyze causes why human being feels discomfortable to a user interface and classify it in seven factors, waiting time, effort, disappointing, senses, messages, unexpected and disturbing. They show that a user interface which has a little bit of such a factor is effective to prevent users from sending e-mail to an incorrect address.

VI. Conclusion

We have proposed the system that compels users to input their schedules by using unpleasant notification. Our preliminary experimental result shows our system shortens the reaction time of the users to input. Future works include we conducts more detailed, long term experimental results, and to change the order of notification methods in each steps and in each users.

References

[1] A Schedule Arrangement Service: Chosuke, http://http://chosuke.rumix.jp/
[2] Kahneman, Daniel, and Amos Tversky ”Prospect Theory: An Analysis of Decision under Risk”, Econometrica, XLVII, 263-291, 1979.
[3] Ian Ayres, Carrots and Sticks: Unlock the Power of Incentives to Get Things Done, Bantam, 2010
[4] stickK - Changes Starts Now, http://www.stickk.com
[5] Hiromi OikawaYasuhiro HujiwaraYuko Murayama, Causal structural model for an interface causing discomfort, IPSJ symposium series, Vol.2007, No.10, pp. 355-360, 2007
[6] Haruka MurayamaYasuhiro HujiwaraYuko Murayama, An implementation of an interface causing discomfort for awareness of risks and threats, IPSJ symposium series, Vol.2009, No.4, pp .141-142, 2009

978-1-4799-2845-3/13/$31.00 ©2013 IEEE