Balancing Awareness and Interruption: Investigation of Notification Deferral Policies

Eric Horvitz, Johnson Apacible, and Muru Subramani

Microsoft Research, One Microsoft Way
Redmond, Washington 98052, USA
{Horvitz, Johnsona, Murus}@microsoft.com

Abstract. We review experiments with bounded deferral, a method aimed at reducing the disruptiveness of incoming messages and alerts in return for bounded delays in receiving information. Bounded deferral provides users with a means for balancing awareness about potentially urgent information with the cost of interruption.

1 Introduction

The increasing dependence on computers for communication has made typical computing environments disruptive places to work. We investigate the promise of a method named *bounded deferral* [3] for providing calmer computing environments. Bounded deferral captures a simple but powerful idea: If users are busy when an alert arrives, they may be willing to wait some prespecified maximum amount of time before being informed about the alert, so as to minimize interruptions in return for a relatively small cost of delayed awareness. With the approach, messages are delivered when the user transitions to a non-busy state, if such a transition occurs before the maximal deferral time. Should the user remain busy, the alert is guaranteed to be delivered at the maximal deferral time. We now review several exploratory studies that probe the potential value of bounded deferral.

2 Studies of Bounded Deferral

We shall now investigate properties of bounded-deferral policies with three studies.

2.1 Analysis of Data from Interruption Workbench Studies

To explore the promise of bounded deferral for reducing the disruptiveness of incoming notifications, we initially examined data that had been collected as part of research with the Interruption Workbench [2]. The two subjects in the study used a tagging tool to label situations captured in a recording of 5 hours of their office activities as either being high, medium, or low cost of interruption states. The video captured the details of work on their computer screens and surrounding office environment. Participant 1, a program manager, spent .20, .61, and .18 of the total time in high, medium, and low cost states respectively, and remained in a busy state for a mean time of 21 seconds before transitioning into a lower cost state. Participant 2, a software developer, spent .29, .48, and .23 of the total time in high, medium, and low cost states, and remained in a high cost state for a mean time of 202 seconds,

L. Ardissono, P. Brna, and A. Mitrovic (Eds.): UM 2005, LNAI 3538, pp. 433 – 437, 2005.

before transitioning into a lower cost state. Assuming that notifications arriving during high costs states come at random times during these periods, we found that Participant 1 would transition into a lower cost state in a mean time of 11 seconds after the arrival of an alert. Participant 2 would transition into a lower cost state at a mean time of 101 seconds after the arrival of an alert. Thus, we found that for these two subjects, allowing a relatively small, bounded deferral on the delivery of messages could significantly minimize disruptions.

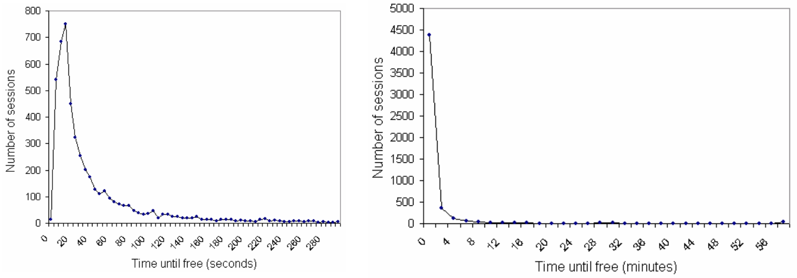


Fig. 1. Left: Distribution of the durations of busy situations for 113 users over three business days. Right: A longer-duration view of the distribution of durations of busy situations

2.2 Analysis of Data from Busy-Context Tool

For the second study, we analyzed data from users who had defined a set of deterministic policies about their availability with a prototype called the Busy Context tool. We had initially fielded the tool within our organization within an application for routing real-time communications based on a user's interruptability. The tool allows users to define when they are busy with rules, and to define available as the complement. Observations include computer activities and conversation near a user's computer. The policies work in conjunction with an event-sensing system that monitors computer activity and that compiles a time-stamped event stream in a computer log. The log can be uploaded continually to a server for studies of user activity. The conversation detector appeared to perform reliably in tests, discriminating typical noise in an office environment from conversations.

We investigated the busy versus free situations for 113 users for several weeks to months at the time of the study. The users included 42 program managers, 25 software developers, 10 administrators, 7 midlevel managers, 2 senior managers, 4 people in sales and marketing, 19 software testers, and 4 research scientists. The participants granted us access to their busy/free definition settings and to their free and busy states. Both the settings and the states were monitored via a server. We analyzed data collected over three sequential business days between 10am and 4pm when users were active at their desktops. We collected 4,803 busy situations. The graph at the left side of Fig. 1 shows the distribution over durations of the monitored busy sessions for the participants. The mean duration of the busy sessions was found to be 43.12 seconds with a standard deviation of 51.79 seconds. The graph on the right side shows a view of the data zoomed out to one hour. The data shows that a great majority of busy situations transition to free situations within 1 to 2 minutes. The results underscore the potential value of bounded deferral. If users associate being interrupted during a busy state with a significantly higher cost of interruption than being interrupted during a free time, then simple bounded deferral strategies can provide great value. We note that the value provided by bounded deferral is likely to be greater than suggested by the results in Fig. 2 as these represent capture the time until transition from the beginning of busy states to the start of free states. This is an upper bound on the times that messages will have to wait as messages arriving while a user is busy can arrive at any time during the busy context.

2.3 A Case Study of Bounded Deferral for Email

We recruited two participants at our organization for a case study with the use of bounded deferral in email alerting. Participant 1 is a program manager and participant 2 is a software developer lead. The participants agreed to share out their email stores for the study. Both participants had been using the Busy Context system in a prototype for routing phone calls, and had also been using an email triage system, called Priorities, that our team had developed in earlier research [1].[1] Priorities assigns scores to each incoming email based on a classifier learned via supervised or unsupervised learning. At run-time, incoming messages are assigned an urgency score. The approximate classification provided by the tool was sufficient for the purposes of our study, which was focused on the overall experience with using bounded deferral with such a system, rather than on a study of the triaging tool.

The messages for each participant were segmented into low, medium, and high urgency messages via the mappings to these categories that the user had made within the email classification system. We asked users to explore the assignments and to note misclassifications by the automated system for messages received during the work week. Although the classifier was judged to perform well, a small percentage of messages were viewed as misclassifications. Rather than use a cleaned data set, we carried out our analyses with the raw output of the automated triage system so as to get a sense for the experience with extending an existing email triage system with bounded deferral policies. We considered all email messages for the two participants for a business week, from 9am to 5pm. Junk email was removed from consideration.

After collecting the data, we aligned the busy and free states of these users, as collected by the monitoring of the user busy states via the log provided from the Busy Context tool, with the receipt time of the incoming mail and considered the number of total email messages that arrived while the user was in a busy state. Assuming a system that would alert users for all incoming email, we considered how different message deferrals would influence the number of alerts while they were in a busy state. We identified, for each deferral setting, the likelihood that messages would disrupt users, based on the alignment of incoming messages and the user's free and busy states. Fig. 2 shows the likelihood that each user would transition to a free situation if currently busy, starting at the beginning of the busy state. These curves were computed by noting all durations of busy sessions as we did in the prior analysis, but normalizing the total number of sessions to 1.0. As indicated by the graphs, both users transition from being busy to being free within two minutes, with the first participant transitioning to free with a 0.5 probability within approximately 25 seconds and the other to a 0.5 probability of transitioning within 50 seconds.

[1] Priorities is the ancestor of Microsoft's Outlook Mobile Manager, an email triaging product.

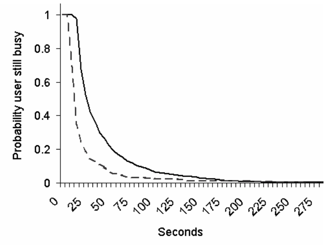


Fig. 2. Probability distribution of the time of transition of participants from busy to free situations (solid curve: participant 1; broken curve: participant 2)

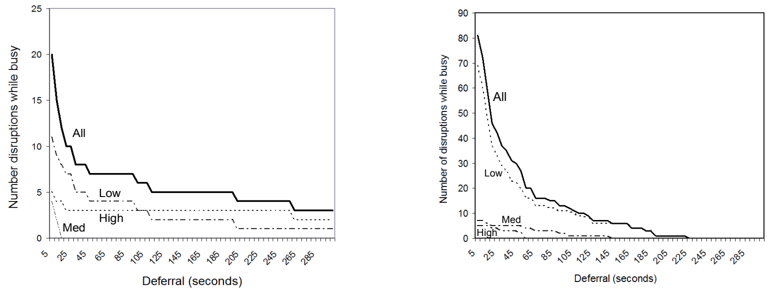
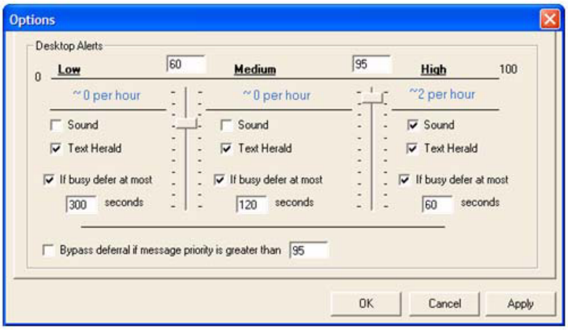


Fig. 3. Number of disruptions for incoming messages when busy as a function of increasing amounts of deferral for participant 1 (left) and participant 2 (right). (Solid curve: all incoming email; Broken curves: email assigned low, medium, and high urgencies by a classifier)

Moving onto an exploration of the value of bounded deferral in the real-context of email, we first explored how often a user would have been alerted with alerts about incoming email during busy times, at different maximal deferrals, for all of their incoming messages. The solid curves in the graphs in Fig. 3 show, for each user, the diminishment of alerts during busy times with increasing deferral times, for all of their incoming email. The broken curves represent the numbers of alerts associated with the arrival of messages assigned high, medium, and low urgency scores. In the all-email condition, a deferral of 4 minutes would lead to a diminishment of alerts during busy times from 20 to 4 alerts for participant 1 and a diminishment of 80 to zero alerts for participant 2. Turning to the curves for the different classes of urgency, for participant 2, a deferral of a little more than 1 minute for email classified as high urgency email would suppress all alerts by important email during busy times with only a small delay in transmission of information to the user. A 3 minute deferral for the user would suppress all medium urgency alerts, and a delay of transmitting low urgency email of 4 minutes would suppress all low-urgency email for a likely tolerable wait to see such alerts. We note that the results are worst-case analyses, as we consider the durations of busy states as starting when the users became busy. Based on the results of our study, we extended the Priorities email triage system with bounded deferral and bounded deferral controls and forecasts. A view of the controls for defining email as low, medium, and high urgency and for setting different deferral policies for the different classes of email is displayed in Fig. 4.



Desktop Alerts

0 Low [60] Medium [95] High 100

Low: ~0 per hour
□ Sound
☑ Text Herald
☑ If busy defer at most 300 seconds

Medium: ~0 per hour
□ Sound
☑ Text Herald
☑ If busy defer at most 120 seconds

High: ~2 per hour
☑ Sound
☑ Text Herald
☑ If busy defer at most 60 seconds

□ Bypass deferral if message priority is greater than 95

Fig. 4. Controls for a tool for specifying bounded-deferral policies for guiding alerting by the Priorities email triage system. Forecasts of the numbers of interruptions per hour for each class of email are displayed above each message-urgency column

Users can use sliders to define how urgency scores for email provided by the classifier are mapped to three urgency classes. Users can also change the way that alerts are rendered or suppress alerting completely by class. To provide users with feedback about the influence of deferrals and mappings of automatic assignments of urgency to email, the system provides a view of forecasts of the number of alerts that they can expect within each category over a workday. The forecasts are computed by maintaining a log of incoming email and its scores, as well as a log of a user's busy states, and making an assumption that recent history is an indicator of the near future. We shall be studying user experiences with this bounded deferral prototype.

3 Summary

We presented studies of bounded-deferral policies. Bounded deferral and variants provide a framework for designing notification systems that take as inputs assessments about a user's willingness to trade off delays in information delivery for reductions in disruption during times that they consider busy situations. We believe that bounded-deferral policies show promise for quieting the noisy chatter of incoming alerts, while allowing people to stay aware of important information.

References

1. Horvitz, E., Jacobs, A. and Hovel, D. (1999). Attention-Sensitive Alerting. In: Proceedings of the Fifteenth Conference on Uncertainty and Artificial Intelligence, Stockholm, Sweden, July 1999, pp. 305-313. Morgan Kaufmann: San Francisco.
2. Horvitz, E. and Apacible, J. (2003). Learning and Reasoning about Interruption, *Proceedings of ICMI 2003*, pp. 20-27.
3. Horvitz, E., Kadie, C. M., Paek, T., and Hovel, D., (2003). Models of Attention in Computing and Communications: From Principles to Applications, Communications of the ACM 46(3):52-59, March 2003.