BusyBody: Creating and Fielding Personalized Models of the Cost of Interruption

Eric Horvitz
Microsoft Research
One Microsoft Way
Redmond, Washington USA
425-706-2127
Horvitz@microsoft.com

Paul Koch
Microsoft Research
One Microsoft Way
Redmond, Washington USA
425-706-2127
Paulkoch@microsoft.com

Johnson Apacible
Microsoft Research
One Microsoft Way
Redmond, Washington USA
425-706-2127
Johnsona@microsoft.com

ABSTRACT
Interest has been growing in opportunities to build and deploy statistical models that can infer a computer user's current interruptability from computer activity and relevant contextual information. We describe a system that intermittently asks users to assess their perceived interruptability during a training phase and that builds decision-theoretic models with the ability to predict the cost of interrupting the user. The models are used at run-time to compute the expected cost of interruptions, providing a mediator for incoming notifications, based on a consideration of a user's current and recent history of computer activity, meeting status, location, time of day, and whether a conversation is detected.

Categories and Subject Descriptors
H1.2. Models and principles: User/Machine Systems

General Terms
Cost of interruption, notification systems, models of attention

1. INTRODUCTION
Efforts over the last several years have demonstrated that relatively accurate models can be constructed for predicting the interruptability of users from sensed activity [3,5,6]. Such models promise to put in the hands of people tools for building personal agents that have the ability to mediate if, when, and how notifications and real-time communications should be relayed to them. For example, in prior work on the Notification Platform project, models of the expected cost of interruption, trained by users in an offline setting, are coupled with models that assign measures of urgency to communications, and fielded within a Web-service architecture [7]. The system deliberates about the cost of transmitting alerts to users in different ways, based on a sensing of desktop activity, calendar appointment data, head pose, and nearby conversation. In that work, models were trained with a system called the Interruption Workbench [6], which synchronizes videotaped sessions of users at work with a recorded event stream, and provides users with a tool that allows the users to tag different points of time in the captured sessions as being associated with costs of interruption. Bayesian network models are then generated from the case library of tagged data. The models provide at run time a probability distribution over the cost of interruption based on sensed events. This probability distribution is then combined with a decision-theoretic policy to generate an expected cost of interruption. Studies of models constructed with the Interruption Workbench explored the relative value and substitutability of sensors in model ablation studies. In work related to the Information Workbench, efforts were made on the Coordinate project to build models of the cost of interruption associated with meetings, based on a collection of training data from users and multiple properties from online calendar data [5]. In recent work related to the Interruption Workbench and to the research reported in this paper, users were prompted while working to record their current interruptability via audio annotation [3]. Data was logged about several desktop features, conversation status, and the configuration of user's office doors. Models were constructed offline to predict user's interruptabilities, based on the sensed data.

In this paper, we present a software component that provides an integrated, onboard supervised learning and inference system, named BusyBody. BusyBody is an evolution of heavier-weight offline training systems packaged into a self-contained system. During a training phase, BusyBody intermittently engages users via a pop-up *busy palette*, heralded with an audio chime. The palette allows users to assess their current cost of interruption efficiently. In the background, a rich stream of desktop events is logged continuously. These events, along with information drawn from the user's calendar, wireless signals, and an onboard conversation detector, are combined with the self-assessments to build a case library. BusyBody trains and periodically re-trains Bayesian network models that provide real-time inferences about the cost of notification. The models are linked to programming interfaces that allow other components, such as notification systems to access the expected cost of interruption. BusyBody can be instructed to execute either entirely on a user's personal computer, or to alternatively package the information locally and to communicate its logs to a server when network connections become available. The use of a central server enables the construction of models that consider activity on multiple machines that the user may use at the same or different locations.

In the rest of the paper, we shall review background on the cost of interruption, describe the sensing and learning infrastructure, and report studies of four users who participated in a study.

2. COST OF INTERRUPTION

We have worked to endow computing systems with an understanding of a user's workload, and overall sense of interruptability. Efforts to build machinery for assessing interruptability have included user studies, the formulation of models of attention and cognitive load, the construction of real-time sensing and reasoning platforms, and the development of applications such as agents for mediating communications and messaging, the focus the work reported here. Beyond research on building models to predict the interruptability of users, researchers have carried out user studies to elucidate the costs of interrupting users in different ways while they are performing different tasks in office settings [2,4,8].

In fielding machinery for deliberating about the mediation of communications, based on inferences about a user's interruptability, we have taken a decision-analytic approach. Before moving ahead to details about BusyBody, we shall provide brief background on this approach, highlighting the semantics of the expected cost of interruption.

We seek to infer the cost C assigned by users of being interrupted by different types of disruptions D conditioned on being in particular states S, $C(D_i,S_j)$. In a messaging setting, such costs can be assessed using decision-analytic assessment techniques such as the *willingness to pay* in dollars to avoid the disruptive component of notifications. Predictive models of user states constructed with training data generate probability distributions over states of interruptability. Thus, we invoke the principles of expected utility decision making to compute the *expected* cost of disruptions under uncertainty in taking mediation actions.

In our prior work, we assessed the costs of disruption as the *willingness to pay* to avoid the negative aspects of a disruption in dollars for alerts of different kinds disrupting the users in different contexts [6]. Willingness to pay to avoid the negative aspect of outcomes has been used as an assessment tool for several decades in decision analyses as applied to such fields as medical decision making. Given a set of dollar values that users assess that they are willing to pay to avoid different kinds of disruptions, and a probability distribution inferred over the state of a user, we compute the expected cost of interruption (ECI) by summing over the costs, weighted by the likelihood of each state, given sensor observations. That is, the ECI is,

$$ECI = \sum_j p(S_j \mid E) C(D_i, S_j) \quad (1)$$

where $p(S_j|E)$ is the probability of the state, conditioned on observational evidence E.

In the general case, we can consider states S as representing a variety of user situations and perform detailed assessments of costs associated with these states, and consider a set of specific kinds of disruptions. In related work, we considered a range of user situations such as focused creative activity, browsing lightweight activity, conversation with a colleague, private/personal time, etc. [7]. However, as our research progressed to deploying systems that users may wish to personalize with ease, we have found we could ease the assessment task by defining a *state of interruptability*, I, and to build models that directly infer this state. With this approach, we ask users to directly assess the cost of being interrupted when they are in different states of interruptability, spanning a spectrum that they define. Users assess the costs for each type of disruption for each of a potentially small number of interruptability states. We can further simplify Equation 1 by fixing D to be a standard disruption, such as that associated with receiving an instant message or email alert. With this assumption, we simply ask users to assign a scalar value of cost to receiving alerts for messages when they are in different states of interruptability. Given uncertain inferences about the state of interruptability, we compute the expected cost as,

$$ECI = \sum_i p(I_i \mid E) C(I_i) \quad (2)$$

The formulation can be further simplified by asking users to asses a small number of states of interruptability, such as whether they are in a state of low or high cost of interruption, and to allow them to define these variables and then map the states to costs.

An expected cost of interruption enables developers to build mediation systems that perform simple cost-benefit analysis, for example, providing a simple slider control for setting a threshold on a computed expected cost of interruption at which messages will lead to desktop alerts versus to a journaling of the information for later review. The behavior of such a system can be inspected and tuned by users via moving the slider up and down.

3. EVENTS, FEEDBACK, AND LEARNING

We now turn to the challenges of collecting training data and constructing of predictive models in BusyBody. BusyBody's event-monitoring infrastructure borrows software components from our earlier projects on cost of interruption and availability forecasting. Specifically, we integrated into BusyBody event-sensing and logging components from those used in the Notification Platform and Coordinate systems.

Event-monitoring methods developed for Notification Platform provide a rich infrastructure for detecting and logging desktop activities, including both low- and high-level events. Low-level events include states such as whether the user is typing, moving or clicking with the mouse, and the current focus and recent history of activity users have with applications and window titles (capturing activity within specific subwindows of applications, in addition to file names and urls). Higher-level events include the timing and pattern of switching among applications and window titles (*e.g.*, how many times the user has switched applications or window titles within different time horizons, and how many unique windows or applications have been visited). We also record the total time that a user has been inactive or active within an activity-time tolerance that allows for brief pauses in activity.

Beyond computing events, we record several classes of contextual variables capturing states beyond the computer. We note that the time of day and day of week. We also consider, via the Microsoft Outlook application, whether a meeting exists, and was marked as tentative or accepted as confirmed by the user. BusyBody includes a conversation detection system, with a module developed at our laboratory that detects acoustical energy in the audio spectrum in the human-voice range. The detector logs whether a user is speaking or last spoke at the time of assessment. If the user is speaking, the system notes in the log how long the user had been speaking (beyond a small silence duration tolerance). The system also logs, via wireless signals detected during sessions, either known (previously registered) or unknown locations. Finally the log includes the name of the computer being

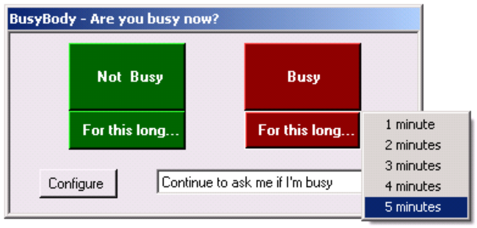

Figure 1. Busy palette, showing selection of optional drop down list for assessing duration of current busy state.

used or that was last used if there is inactivity up until the probe, when it is operating with a server that consider a user's different computers.

When users activate BusyBody's training mode, they grant the system permission to interrupt them intermittently to ask about their current cost of interruption. Users can set experience-sampling parameters in BusyBody's control panel that is accessible in the system tray of the Windows shell. Users can set their preferences about the number of tolerated assessments per hour. Users can also swiftly put the assessment function "to sleep" for an hour, a day, or until the system is reactivated. BusyBody attempts to sample smoothly across time, but randomizes the exact time it will appear, while maintaining its obligation to sample the specified number of times each hour. Beyond waiting for the probes, users can manually invoke the palette to instruct the system about the cost of interruption associated with a particular setting.

We have experimented with assessment palettes of varying complexity. An early version of BusyBody employed a three-state assessment, asking users if they were in a situation, associated with low, medium, or high cost of interruption. For the studies reported in this paper, we employed a two-state assessment palette that users participating in our studies reported was easier to assess. Beyond simplicity, recent related work has provided support for segmenting interruptability into highly noninterruptible versus other situations [3].

Figure 1 shows the two-state busy palette. The notification appears in a location on the display set by the user with a short-lived, audio herald of a harp being strummed gently. Users can immediately dismiss the assessment tool by clicking on the *Not Busy* or *Busy* buttons, colored green and red, respectively. Users also can tell the system that they have been in a low or high cost of interruption for extended periods of time by clicking on the smaller "For this long…" buttons beneath the main targets, accessing a drop down box containing a list of durations. If a user does not click on the assessment within 60 seconds, the palette disappears, and the case is recorded as unanswered.

Rather than serving as an end-to-end application like its ancestor Notification Platform, BusyBody was initially designed as a piece of core infrastructure for building notification-throttling applications. If the ultimate mediation system built on top of BusyBody is one that provides users with an adjustable slider to specify thresholds on the cost of interruption at which different classes of real-time notifications will be suppressed, there is no reason to require that users assess specific costs, but rather to simply use a simple, linear cost function assumed by default. However, the system also provides a utility-assessment component that allows users to assign specific values to the cost of being interrupted in different states, enabling more

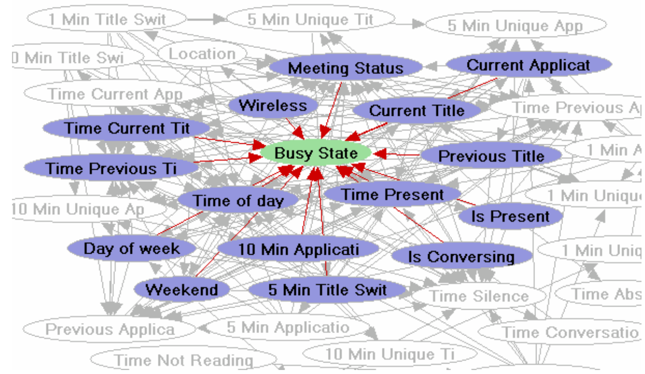

Figure 2. Portion of Bayesian network built by BusyBody for participant P1 highlighting key influencing variables.

sophisticated cost-benefit analysis of message types given that users are in different states of interruptability. Thus, BusyBody can support a spectrum of end-application sophistication.

Given the assessments and logs of a stream of high-level and low level computational events and contextual evidence mentioned earlier, the system builds a library of cases, and then employs a Bayesian learning procedure, employing graph structure search [2]. This process generates a personalized Bayesian network model for the cost of interruption which can be used for making real-time predictions.

4. STUDY AND RESULTS

We initially fielded the application to several participants at our organization. Although we generally wished users to come up with their own definitions of "busy" versus "not busy" in assessing the two-state model, we provided general guidance; users were asked to tag their situation as busy when they believed that would prefer to temporarily suppress the receipt of messages that might be of importance to them. Probes that were not answered while the user was using either using the computer or engaged in conversation were marked as busy states.

We will now review the predictive accuracies of models constructed by the BusyBody systems of four participants after several weeks of running BusyBody. Participant P1 is a program manager who performs a great deal of communication daily and manages multiple projects, involving several teams of people. P2 is a software developer team lead. P3 and P4 are software developers who are typically tightly focused on specific programming tasks. Table 1 shows the number of assessed states and the classification accuracies of models constructed from the cases for each participant. The classification accuracies were computed by dividing the case libraries into training and test sets with an 80/20 split, constructed by randomly drawing the subset of training cases from participants' case libraries.

Beyond studies of the predictive accuracy of the models, the learned Bayesian networks were inspected, in pursuit of insights about key variables influencing the participants' reported busy states. A portion of the learned Bayesian network for participant P1, highlighting key discriminatory variables, is displayed in Figure 2. For this participant, key influencing variables, in order of their probabilistic influence on the user's assessment of being busy versus not busy, include the number of window titles that had been visited in the last 5 minutes, whether the user was active on a computer, whether the user was engaged in conversation, the day of week, the titles of the windows currently and previously in

Table 1. Classification accuracies.

Participant	Training Cases	Accuracy
Participant 1	2365	0.87
Participant 2	789	0.70
Participant 3	1449	0.85
Participant 4	470	0.71

focus, the application being used, the duration of the current session on the computer up to the assessment time, the time of day, the user's location, as determined by wireless signals, and meeting status. Table 2 shows the top ten discriminating variables for each of the participants, sorted by a measure of probabilistic influence generated by the model construction procedure. We note that participant P3 did not have the conversation-detection component turned on. A number of variables and variable classes show strong influence across the participants, including durations of current and previous applications or window titles, and rates of shifting among applications or windows at different time horizons in advance of the assessment.

In interviews following the training period, participants reported that they found the intermittent probes of BusyBody's assessment palette somewhat annoying but they appeared to maintain a friendly attitude toward the inquisitive system that had been curiously nosing its way onto their desktops during training.

Figure 3. Real-time inference with BusyBody models showing probability distribution and expected cost of interruption for a user.

At run time, the models learned by BusyBody are used to reason from the events and states sensed by the system to provide a probability distribution over states of interruptability. The system computes an expected cost of interruption, using the probability distribution and assessed costs of interruption for the high and low states of cost of interruption. Figure 3 shows a view of the inferred expected cost of interruption for a participant provided by BusyBody instrumentation for the two-state model. In this case, the user has assigned a one-dollar cost to being interrupted when busy and no cost to being interrupted when free.

5. DIRECTIONS

We found the experience and results with preliminary trials with BusyBody to be promising. We are planning to field BusyBody to a significantly larger group of participants and look forward to analyzing the results from the wider-scale study. We are studying several practical issues with fielding the system, including the immediate use of the system without training, or with training that occurs over time. For the former challenge, we have been pursuing an understanding of the predictive power of models built from data from one user or from groups of users to predict the interruptability of others. For the latter issue, we have implemented a model-averaging methodology that combines the results of a seed model and a personalized model, with a weighting that shifts toward the personalized model as the model becomes more competent as it is populated with increasing numbers of cases. Finally, we are working to better understand how the interfaces to learned BusyBody models might be extended and enriched to enable application developers to harness real-time inferences about a user's interruptability.

Table 2. Key influencing variables for participants.

Participant 1	Participant 2	Participant 3	Participant 4
Title shifts 5min	App. focus	App. focus	App. focus
Active on machine	Day of week	Title focus	Day of week
Conversation	Previous title	App. shifts 5min	Duration app. focus
Day of week	Title shifts 5min	Title shifts 10min	Title shifts 1min
Title focus	Duration of silence	Day of week	Title focus
Prev. title focus	Title shifts 1min	Duration title focus	Duration of session
App. focus	App. shifts 10min	Title shifts 5min	Active on machine
Duration of session	Time of day	Time of day	App. shifts 5min
Time of day	Previous app.	App. shifts 1min	Time of day
Location	Duration of session	Duration app. focus	App. shifts 1min

We believe that the BusyBody approach represents a step forward in the vision of deploying systems that are aware of users' workloads. To date, systems that reason about a user's workload have largely remained inside the walls of research laboratories. We believe that a descendant of BusyBody will break into the real world to provide value to users.

6. REFERENCES

[1] Chickering, D.M., Heckerman, D. and Meek, C. (1997). A Bayesian approach to learning Bayesian networks with local structure. In *Proceedings of UAI 1997*, pp. 80-89.

[2] Cutrell, E., Czerwinski, M. and Horvitz, E. (2001). Notification, Disruption, and Memory: Effects of Messaging Interruptions on Memory and Performance, *Proceedings of Interact 2001*, pp. 263-269.

[3] Fogarty, J., Hudson, S.E., and Lai, J. (2004). Examining the Robustness of Sensor-Based Statistical Models of Human Interruptability, Proceedings of CHI 2004.

[4] Gillie, T. and Broadbent, D. (1989). What makes interruptions disruptive? A study of length, similarity and complexity *Psychological Research*, 50, 243-250..

[5] Horvitz, E. Koch, P., Kadie, C.M. Jacobs, A. (2002). Coordinate: Probabilistic Forecasting of Presence and Availability. *Proceedings UAI 2002*, pp. 224-233.

[6] Horvitz, E. and Apacible, J. (2003) Learning and Reasoning about Interruption, *Proceedings of ICMI 2003*, pp. 20-27.

[7] Horvitz, E., Kadie, C. M., Paek, T., and D. Hovel. (2003). Models of Attention in Computing and Communications: From Principles to Applications, *Communications of the ACM* 46(3):52-59.

[8] McFarlane, D. (1999). Coordinating the interruption of people in human-computer interaction. *Proceedings of Interact '99*, pp. 295-303.