Interruption Management in Web-based Collaborative Systems

Syavash Nobarany, Mona Haraty, Steve DiPaola, Brian Fisher
School of Interactive Arts and Technology
Simon Fraser University
Surrey, BC V3T 0A3, Canada
{sna35, mha48, bfisher, sdipaola}@sfu.ca

ABSTRACT
Interruption management has received a good deal of attention from researchers, but this work has not had a substantial impact on real-world applications. This paper examines the factors associated with interruption management in web-based systems (a highly restricted class of environments) and presents an adaptive interruption management system, Interrupt-Broker, as a component of a web-based visual analytics toolkit. Interrupt-Broker seeks to find the right time for delivering messages in such a way as to minimize the cost of interruptions. These costs include distraction from the primary task and disruption of cognitive processes through premature incorporation of new information in the course of sensemaking (e.g., conformity to a collaborator) in a collaborative analysis and decision-making system. A hybrid deliberative/reactive architecture, inspired by enactive cognition theories, is designed to deal with minimal sensory information gathered from the user interaction to determine users' cognitive state. This architecture implements reinforcement learning and reactive action selection.

Author Keywords
Interruption, messaging, enaction, learning, collaboration.

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION
Computer supported collaborative work (CSCW) has been a prominent research area for decades. Instant messengers are one example of successful computer mediated communication (CMC) tools that are designed for synchronous and asynchronous “different place” communication (based on time/space matrix [10]). Although most CSCW systems support multimedia communication [28], text messaging retains its importance among CMC tools. This is due to the persistence of communication, possibility of error correction, and indexability of chats, all important characteristics of successful CSCW systems [19,21].

Another advantage of text messaging is that users can tolerate delays in textual conversations more easily than other types of communication. Although there have been efforts like [29] for incorporating this feature into other media, none of them have had the same success. This characteristic in particular is vital for interruption management. Interruption management has been one of the frequently addressed issues in human-computer interaction research. There have been efforts showing the negative effects of interruptions and the necessity of managing them [1,8] followed by research on finding rules for interruption management [4]. Several studies have shown the effectiveness of interruption management systems. Some of them are predicting breakpoints [15,16], while others using probabilistic models of interruptibility [12,13,17] by taking advantage of a large number of sensors for detecting different types of user engagement.

Recent research in cognitive science suggests another approach to interruption management in which, rather than trying to predict the cognitive state of the user by monitoring his environment, we instead monitor the users’ behavior as an indicator of his cognitive state. According to enactive cognition theories, the user’s state of mind is a complex function of perceptual experience and cognitive task as both unfold over time[25]; therefore instead of increasing the number of sensors to gather information about user environment, we might be able to focus on sensor data that enables us to fix the terms of a cognitive model that will give us some information about the user's cognitive processes. The problem is that the most informative sensory systems such as functional magnetic resonance imaging (fMRI) and magneto-encephalography (MEG) are not deployable in a workspace environment due to their large size and heavy weight [23] and sensitivity to
electrical and magnetic interference; less invasive systems such as electroencephalography and pupil dilation are not applicable to a web-based system in which users cannot be instrumented in this way. The only sensory information that is always available is the user’s interaction. Although the combination of physiological and behavioral sensory information can help in achieving a better estimate of user cognitive state, some of the simple interactions such as speed of mouse motion can be helpful in measuring task difficulty, stress and arousal [14]. Also the waveform of clicks is another measure of task difficulty [24].

Given out goal of building a web-based application, we must predict user interruptibility based on data generated by a conventional user interface and we hypothesize that there is a usable correlation between user behavior and interruptibility.

COLLABORATION AND INTERRUPTION

Unscheduled communications are inevitable in collaborative systems. Even relevant messages may impact concentration on the task if they arrive at an inopportune time. One of the most desirable types of communication in a collaborative system are comments or annotations associated with an artifact which the user is working on. If these messages arrive before the user has processed the artifact themselves, social conformity [3] will drive them towards premature agreement with the majority. In the context of recommender systems, predictions of a user’s opinion about a movie have significant effects on his rating [7]. This problem becomes more important in collaborative visual analytic systems where analysts are expected to discover unexpected patterns in the visualization.

Other types of messages, may lead to user distraction and therefore decreasing user performance. Based on [20] studies have suggested if-then rules for distinguishing time intervals when users are less sensitive to interruptions. Although these findings are useful and effective rule-based prototypes have been developed [22], the lack of a unified theory of cognition can lead to conflicting rules from different studies, e.g. on the effect of notification delivery in early stages of task life-cycle [9].

Predicting user interruptibility state

The main difference of Interrupt-Broker with previous systems is that it is informed by new theories of cognitive task performance and user characteristics. Interrupt-Broker is designed to behave rationally in web-based applications, which means that it must rely on the minimal sensory information available through user interaction. User’s visible behavior for Interrupt-Broker consists of first and second derivatives of mouse movements and clicks, opening and closing workspace components and the type of component that user is working with. Before running a full-size user study, further preliminary study is required to find a suitable state-definition variable set. Considering the small amount of sensory information available, ideas from enactive/embodied cognition are applied to help the agent’s decision making process.

Embodied cognition

The enactive cognition approach focuses on the primary role of interaction in the cognitive processes of intelligent systems. The world of an agent is brought forth by the structural coupling of the agent and its environment [26]. For example, Kirsh & Maglio studied the performance improvements that can be achieved by the physical actions that uncover information that are hard to compute mentally, known as epistemic actions [18]. This perspective leads to considering interaction with the environment as an active component of cognition, thus avoiding redundantly modeling and creating representations of the environment [5,6].

Reactive robots serve as an example of enactive interfaces that are continuously in interaction with the real world [2], however the design of enactive soft-bots is often overlooked [11]. Interrupt-Broker demonstrates an enactive interface design which uses epistemic action and reactive behavior, acquiring knowledge by experiencing and doing (inherent to reinforcement learning) and that revises its perception process based on experience.

Interrupt-Broker

Design

Interrupt-Broker is designed as a built-in messenger in a web-based collaborative visual analytics toolkit. It uses a hybrid reactive/deliberative architecture which is basically consists of two layers: reinforcement learning which is performed server-side and the reactive action selection in client-side. In reinforcement learning terms, the user’s reactions are considered as rewards and punishments that Interrupt-Broker receives and user behavior shapes the state of the world. The choice of action for the reinforcement learning algorithm is: waiting, delivering the most urgent message, or expanding or shrinking the sensing intervals.

The reactive behavior of Interrupt-Broker is designed to make sure that the system does not interrupt any operations (but not tasks). It also allows for delivering messages that are associated with interactions or information artifacts such as annotations which server-side module will miss delivery of due to the client-server communication delays. Similar process happens in human body, where some messages like touching a hot object can be responded to, immediately, with a reflex response, via the spinal cord instead of the brain. Another reactive behavior happens when an instant message arrives or a message reaches a state of urgency. These reactive behaviors are designed to fine-tune behaviors that are generated by the reinforcement learning module and prevent the system from behaving poorly, for example by delivering messages before approaching their deadlines and not breaking an operation (drag, move…). All of these behaviors help Interrupt-
Broker to show basic rational behaviors even during the initial steps of the learning process.

There will be situations where the user’s natural behavior does not provide enough information for Interrupt-Broker to make proper decisions (e.g. no interaction). In that case, Interrupt-Broker performs an epistemic action and uses the environment (user reactions) to predict user’s state.

Design decisions in developing an interruption manager are highly dependent on the context in which the system is designed for. For example in critical environment such as cockpit or surgery environment, even the smallest interruptions should be handled seriously, because small distractions may lead to disasters. But in a web-based collaborative system, although small distractions like the one that is caused by the epistemic action are not costless, but their costs are incomparable with interruptions that impose a secondary task or contain information that may distract the flow of analysis.

Prototype
Interrupt-Broker is implemented using Flex SDK for user interface design (figure 1) and reactive behavior and J2EE for the learning process. Interrupt-Broker has a priority queue of messages based on their deadline and periodically (based on its sensing interval) senses the state of the user and chooses one of these actions: delivering the message at top of the priority queue, changing the sensing interval or waiting. This decision is made based on its previous experiences through reinforcement learning process. For the states that are being experienced for the first time, experiences in similar states are used to predict the value of each action.

Prototype
Interrupt-Broker is implemented using Flex SDK for user interface design (figure 1) and reactive behavior and J2EE for the learning process. Interrupt-Broker has a priority queue of messages based on their deadline and periodically (based on its sensing interval) senses the state of the user and chooses one of these actions: delivering the message at top of the priority queue, changing the sensing interval or waiting. This decision is made based on its previous experiences through reinforcement learning process. For the states that are being experienced for the first time, experiences in similar states are used to predict the value of each action.

Fig. 1. Sample message

When an opportunity for delivery is detected, the message is sent to the client-side module which as the first pause asks the user if he wants to see the message. The permission dialog (figure 2) appears under the cursor to make it fast and easy to accept or reject a message. Not providing information about the message makes it impossible to assess the value of reading message [27]; however the majority of the cost associated with a notification is due to the distraction or other side-effects that are caused by the contents of the notification. Ultimately our design decision is based on the context of installing Interrupt-Broker which is a collaborative analysis system in which a great number of interruptions are associated with a secondary task which leads to costly distraction and long resumption lag.

Fig. 2. Permission interface

The reaction of the user (accepting, rejecting or ignoring the question) will be reported to the server-side module as a new experience and the expected value of actions will be updated.

Unlike previous systems, annotations only appear as a reactive response of Interrupt-Broker to the user's behavior, such as moving the cursor for more than a set amount of time near the point that the message is associated with. This will provide enough time for the user to think about graphical patterns or text information that may be of interest to him alone, and to shape his opinion about a certain part of the graph before being affected by others' opinions.

When user is not interacting, Interrupt-Broker performs an epistemic action, where it moves the mouse cursor to the center of the component in focus (figure 3) and decides about the state of user based on the user's reaction. A fast reaction of the user in moving the cursor means that he is focused on the workspace, but ignoring the cursor means that he is not focused and messages can be delivered without distraction.

Finally to make users comfortable with rejecting distracting messages the user may check if there are any new messages by clicking on the icon on the bottom right corner of the screen. Although it was possible to change the appearance of the icon based on the availability of messages, we decided not to do that, because the change in appearance of that icon, might distract user and is considered as a simple notification and knowing the availability of a new message might prevent user from being focused on the task at hand. Again this decision can only make sense when secondary tasks are common and highly costly, which is the case in many web-based collaborative systems.
CONCLUSIONS AND FUTURE WORK

Interrupt-Broker is a prototype of an adaptive web-based interruption manager which is designed to be able to deal with minimal sensory information for determining user interruptibility state using interaction methods inspired by enactive cognition theories.

Upon completion of the preliminary user studies, Interrupt-Broker needs to be incorporated in a functional web-based collaborative visual analytics system to make it possible to perform a comprehensive evaluation of the system in actual applications and with actual users and due to the experience basis of Interrupt-Broker’s learning process, this step is particularly important.

ACKNOWLEDGEMENTS

This work was supported by the Natural Sciences and Engineering Research Council of Canada.

REFERENCES


