

# QnA: Augmenting an Instant Messaging Client to Balance User Responsiveness and Performance

Daniel Avrahami and Scott E. Hudson  
Carnegie Mellon University  
Pittsburgh, PA 15213  
{ nx6, scott.hudson }@cs.cmu.edu

## ABSTRACT

The growing use of Instant Messaging for social and work-related communication has created a situation where incoming messages often become a distraction to users while they are performing important tasks. Staying on task at the expense of responsiveness to IM buddies may portray the users as impolite or even rude. Constantly attending to IM, on the other hand, may prevent users from performing tasks efficiently, leaving them frustrated. In this paper we present a tool that augments a commercial IM client by automatically increasing the salience of incoming messages that may deserve immediate attention, helping users decide whether or not to stay on task.

## Categories and Subject Descriptors

H.5.3 [Information Interfaces and Presentation]: Group and Organization Interfaces—Computer supported cooperative work; K.4.3 [Computers and Society]: Organizational Impacts—Computer-supported collaborative work; J.4 [Social and Behavioral Sciences]: Psychology, Sociology;

## General Terms

Management, Performance, Design.

## Keywords

Instant Messaging, IM, Interruptions, Workflow.

## 1. INTRODUCTION

Instant messaging, or IM, is becoming an increasingly popular conversation medium for both social and work-related communication. IM programs, or clients, facilitate one-on-one communication between a user and their list of contacts, commonly referred to as ‘buddies’, by allowing them to send and receive short textual messages (named instant messages).

Previous research (for example [9,11,13]) has shown that messaging in the workplace has a number of uses and benefits, including opportunistic interactions, broadcasting of information or questions, and negotiation of availability for interaction. Research also shows that users often multitask when using IM [8,11,13]. While many of the benefits of IM come from its near-synchronous nature, it is the asynchrony that allows users to multitask. With computers often permanently connected to the internet, users are able to keep their IM clients running

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CSCW’04, November 6–10, 2004, Chicago, Illinois, USA.

Copyright 2004 ACM 1-58113-810-5/04/0011...\$5.00.

continuously in the background. This means that incoming messages often arrive when the user is engaged with other tasks, possibly in the midst of intensive work. As noted by [14] and [17] it is often the case that time and topic are convenient for the initiator (in this case, the buddy) but not the recipient.

In an attempt to alleviate the problem of IM disrupting work on an important task, or being forced to ignore incoming messages in order to maintain workflow, we have created a small tool, that we call QnA, for automatically alerting users to specific messages that may deserve their attention – in particular to potential questions and answers.

The following scenario illustrates the use of this tool:

## 1.1 Illustration of Use

Jim is in his office preparing a presentation for a meeting that same afternoon. As usual, he is running an IM client with QnA in the background for fast communication with his colleagues. He is missing a few figures and sends an IM to his colleague Bill “did you mean to remove the figure from slide 5”. Bill does not reply and Jim goes back to the presentation. Being short for time, Jim ignores a couple of incoming messages when he notices a QnA notification saying that Bill may be replying to his question. Jim clicks on the notification, bringing the message from Bill to the front. It reads “no, definitely not”. Jim then notices a QnA notification saying that Liz is asking him a question. He clicks on the notification to find a message that reads “do we have a projector?” As Jim is typing his reply, Liz sends another question and Jim modifies his reply. Since Jim was typing, QnA determines it doesn’t need to show a notification for Liz’s second question.

## 2. BACKGROUND

An instant message is regarded as a less intrusive way of interrupting than a phone call or a visit. As noted by [13] it also offers users “plausible deniability”, (that is, the ability to deny presence or receipt of a message, even after having read it.) However, the common alerts associated with incoming messages (the message window opening, sound, and flashing or bouncing icons), even if brief, can easily distract the user and interfere with their work (for discussions on the effects of interruptions on performance see for example, [5,7,10,12,14]).

Being disrupted by message alerts is made worse by the fact that most IM clients have identical alerts for all incoming messages, not taking into account the identity of the sender or the content of the message. In addition, users tend to send many short messages, even when these constitute a single conversational turn. [11] suggest that experienced IM users are more prone towards this behavior. The result is the user being subjected to a large number of alerts, one for each of these short incoming messages.

We now describe a number of strategies available to users for handling the distractions from incoming messages:

[13] report that some users complained about being distracted by alerts while working towards important deadlines. These users reported having to sometimes resort to shutting IM down. As [11] note, however, most IM conversations held in the workplace are work-related. This makes closing the IM client a less desirable strategy.

Changing their online status allows users to indicate to their buddies that they are busy or unavailable. This strategy depends on buddies to recognize and heed this indicator. It also runs the risk that users may forget to reset their status once they are available again, making this indicator unreliable ([2] presented a system that learns the user's work rhythms over time, providing buddies with estimates of the user's online presence).

Next, users can elect to stay on task and ignore, to the best of their ability, the alerts of incoming messages. Different messages are associated with different expectations for levels of responsiveness. These include expectations for a quick response (e.g. – in the message “do you have the figures I need for the meeting?”), a leisurely response (e.g. – “check this out [www.interesting.com](http://www.interesting.com)”), messages that can politely be deferred (e.g. – “busy?”), and messages that do not need a response at all (e.g. – “going to a meeting. ttyl”). Not responding to messages that are associated with expectations of a quick response, may portray the user as impolite, or even rude, and may adversely affect the buddy, if they need information to proceed with their work.

We created QnA to help users identify incoming messages that potentially require a quick response and messages that they are expecting. More specifically, we chose to notify users on incoming questions and answers.

## 2.1 Why Questions and Answers?

Schegloff and Sacks describe in [15] the concept of adjacency pairs in conversation and give question-answer pairs as one type of adjacency pairs. In [4] Clark describes the question-answer pair as the prototype of adjacency pairs:

“Adjacency pairs consist of two ordered utterances, the first and second pair parts, produced by two different speakers. [...] One crucial property is conditional relevance. Given a first pair part, a second pair part is conditionally relevant, that is, relevant and expectable, as the next utterance. Once A has asked the question, it is relevant and expectable for B to answer in the next turn.” (p. 157)

We consider an incoming instant message that contains a question to be representing a first pair part (thus a response from the user is “relevant and expectable”) and an incoming instant message in response to a question to be representing a second pair part (thus the user is likely to be expecting it). If we establish that the user did not attend to these messages for a certain period of time, we notify the user of the pending message, the identity of the sender, and whether the message represents a question, a possible response to a question, or both.

## 3. IMPLEMENTATION

We have implemented QnA as a plug-in for Trillian Pro, a commercial IM client developed by Cerulean Studios [3], and running on Windows operating systems. Like a number of other IM clients, Trillian Pro, and the freeware version Trillian, allow users to use any of the major instant messaging services (AIM, ICQ, MSN, Yahoo!, and IRC) in one application simultaneously.

We chose to use Trillian Pro as it also offers the use and development of dedicated plug-ins through a free Software Development Kit (SDK) giving access to most of the client's functionality. We believe that this, as well as a strong plugin development community, makes Trillian Pro a suitable platform for development and research. Our plugin was written in C and implemented as a Dynamically-Linked-Library (DLL) that is run from inside Trillian Pro. QnA is available for download to Trillian Pro users from the Plugin Development forum on the Trillian website.

### 3.1 Events and Flow Control

QnA uses three internal flags for every buddy the user is sending or receiving messages from. These flags allow QnA to keep track of messages and to determine whether it should present a notification to the user. The flags are: `expectingResponse`, `incomingResponse`, and `incomingQuestion`.

#### 3.1.1 Processing Outgoing Messages

When the user sends an outgoing message to a buddy, QnA scans the message and, using a set of string matching rules, determines whether or not the message is likely to contain a question (For description and discussion of the set of rules used see section 3.2). If it estimates that the message contains a question, it then sets an internal flag called `expectingResponse`, indicating that the user may be expecting a response from this buddy. If QnA estimates that the message does not contain a question, it does nothing.

#### 3.1.2 Processing Incoming Messages

When an incoming message from a buddy is received, QnA first estimates whether the message contains a question using the same

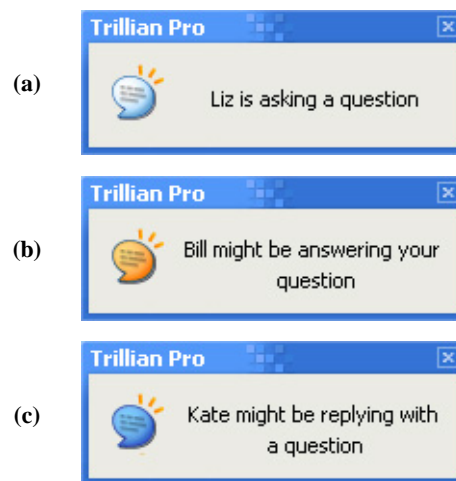


Figure 1. QnA notifications: (a) Question (b) Possible response (c) Question and possible response

string matching rules. If so, it sets an `incomingQuestion` flag, indicating internally that the user might want to respond to this message. It also checks whether or not the `expectingResponse` flag was set for this buddy. If it was, then it is reset, and the `incomingResponse` flag is set instead, indicating that the buddy may have responded to a question.

### 3.1.3 Notifying the User

If either the `incomingQuestion` or `incomingResponse` flags is set (or if both are) then QnA starts a process that waits a certain number of seconds (configurable by the user, the default value is 10 seconds). If, at the end of the wait period, only the `incomingQuestion` flag is set, a small (non-modal) notification similar to the alert shown in figure 1a is presented on the user's screen. If, however, only the `incomingResponse` flag is set, a notification similar to the one shown in figure 1b is presented. If both flags are set the user is presented with an alert similar to the one shown in figure 1c. After the notification is shown, all flags are reset. This is done so that no more than one notification per conversation will be shown every wait period, allowing users to ignore the notifications if they choose to.

If the user clicks on a notification, the message window is opened. If the window is already open, it is brought to the front. If the user does not click on a notification for 10 seconds, it automatically fades out and disappears.

### 3.1.4 Suspending Notifications

Whenever the user is typing a message to a buddy, opening a message window for that buddy, or if the message window is in focus, we assume that the user will have seen any incoming message, and suspend any notifications regarding messages from that buddy. We do so by resetting both the `incomingQuestion` and `incomingResponse` flags. This allows QnA to intercept notifications even if they are already in the wait period.

## 3.2 Identifying Questions

In order to determine whether a message contains a question, we compare the message against a set of string matching rules. We identify the message as a question if any match is found. Note that all matching we perform is case-insensitive. Figure 2 shows a partial list of the rules used. We have also created a set of rules to try and eliminate phrases that should not be considered questions, but that match at least one of the rules (these can be regarded as 'false-positives'). These are mostly questions that serve the purpose of negotiating the availability of the user. Figure 3 shows a few of the rules used. Although this method of identifying questions is not foolproof, it seems to work fairly well in practice.

## 3.3 User Preferences

We allow users to customize the plugin in a number of ways. Users can set the number of seconds that QnA waits before showing a notification. If set to zero, notifications appear instantaneously and no suspension of notifications can occur. Users may also select to be notified only on questions, or only on responses to questions. Finally, users can decide whether notifications should be suspended when typing or when opening the message window. In the future, we plan to allow users to customize the string matching rules.

```
'?' at the end of a line or sentence
 '/' at the end of a line (a common typo for '?')
what (is|are|r|were|does|do|did|should|can)
where (is|are|r|were|does|do|did|should|can)
when (is|are|r|were|does|do|did|should|can)
how (is|are|r|were|does|do|did|should|can)
who (is|are|r|were|does|do|did|should|can)
did(|n't|nt) (i|u|you|he|she|they|we)
do (i|u|you|he|she|they|we)
will (i|u|you|he|she|they|we)
should(|n't|nt) (i|u|you|he|she|they|we)
(are|r) (you|u)
huh
```

**Figure 2. String matching rules used to estimate whether a message contains a question (partial list)**

```
(are|r) (you|u) there
hello?
busy?
how (are|r) (you|u)
```

**Figure 3. String matching rules for messages that should not be considered a question.**

## 4. DISCUSSION

Following our experience using the very first version of QnA, we realized that the presentation of notifications about questions or answers from a buddy should be suspended if the user is already engaged in conversation with that same buddy. Otherwise, we run the risk of constant interference with the already ongoing conversation. We accomplished that in the second version by introducing a delay between the time an incoming message is received and analyzed and the notification of the user. If during the delay we establish that the user is engaged in the conversation with the buddy, we do not show the notifications. We use typing, opening the message window, and a message window being in focus as indicators of engagement in conversation. We specifically chose not to use closing of the message window, as the user might close the window without realizing that a message has been received.

As indicated above, we intentionally allow only one notification per delay period as additional notifications may only distract the users (in particular if they chose to ignore the incoming messages). By aggregating notifications of messages that arrive close to one another, we hope to reduce attention demands without reducing the usefulness of the information.

Identifying questions and answers reliably in instant messages is a challenging task. As noted in [13,16], relaxed grammar and spelling are the norm. Furthermore, instant messages often contain abbreviations. These include abbreviations for single words (for example, 'u' to mean 'you'), or for whole sentences (for example, 'ttyl' to mean 'talk to you later'). The message "r u ready 2 go", for example, needs to be identified as a question.

There are a number of reasons for this. The first is that IM buddies, as opposed to chat or email, are almost always familiar with one another. Users are less concerned about being perceived

as ineloquent, giving priority to sending the message fast. The second reason, and possibly more important one, is the desire to keep the conversation as synchronous as possible. Delaying sending a message to correct spelling or fix grammar can slow the conversation down or even suggest a change in conversation turns. Thus, users may elect to send a message containing a grammatical or spelling error.

Identifying answers to questions reliably in IM can be even harder. The main reason is the multi-threaded nature of IM conversations. As [16] shows, following a multi-threaded conversation can be so hard that it may even confuse the people participating in the conversation.

Researchers in the area of Natural Language and Information Retrieval are working hard to address the problem of identifying questions and matching answers [1,17]. The solutions they propose may indeed be useful for the tool described in this paper. However, as the availability of message persistence can cause users to send many short messages [6], an incoming message may in fact be part of an answer, but not the whole answer. This may prevent the more sophisticated solutions from providing significant improvement.

We believe that notifying the user of the first incoming message following a question, combined with “cautious” notification wording (“X might be replying to your question”), is a reasonable interim solution.

## 5. CONCLUSIONS & FUTURE WORK

This paper has presented QnA, a tool that augments a commercial IM client. By monitoring incoming and outgoing messages QnA allows users to maintain a flow of work by providing salient notifications of incoming messages that may deserve their attention.

While the comments about QnA that we received so far were positive (one user said they thought QnA was “simple, yet useful”), we still need to evaluate whether and how QnA changes the behavior of IM users, in particular when they multitask. In order to do that, we are planning to integrate a mechanism for the collection of anonymous usage data into QnA.

We are also continuously expanding the set of rules for determining if a message contains a question and rules for messages that are not questions. We are currently examining the option to allow users to create custom rules (for example, a user might choose to be notified on a message that contains the string “is dad”).

## 6. ACKNOWLEDGMENTS

Our thanks to Darren Gergle for sending us so many instant messages that we had to come up with, and implement, this solution. This work was funded in part by the National Science Foundation under Grants IIS-0121560, IIS-0325351, and by DARPA.

## 7. REFERENCES

- 1 Agichtein, E., Lawrence S., & Gravano, L. (2001) Learning search engine specific query transformations for question answering. In *Proceedings of the tenth international conference on World Wide Web*, NY: ACM Press, 169-178.

- 2 Begole, J., Tang, J.C., Smith, R.E., and Yankelovich, N. Work rhythms: Analyzing visualizations of awareness histories of distributed groups. In *Proceedings of CSCW '02*, NY: ACM Press, 334-343.
- 3 Cerulean Studios <http://www.trillian.cc>
- 4 Clark, H.H. *Arenas of Language Use*. Chicago & CSLI, 1992.
- 5 Czerwinski M., Cutrell E. and Horvitz E. (2000) Instant messaging and interruption: Influence of task type on performance. In *Proceedings of OZCHI '00*. Academic Press, 356-361.
- 6 Gergle, D., Millen, D., Kraut, R.E., and Fussell, S.R. Persistence Matters: Making the Most of Chat in Tightly-Coupled Work. In *Proceedings of CHI '04*. New York: ACM Press, 431-438.
- 7 Gillie, T., and Broadbent, D. What makes interruptions disruptive? A study of length, similarity and complexity. *Psychological Research*, 50, 1 (1989), 243-250.
- 8 Grinter, R. and Palen, L. Instant Messaging in Teen Live, In *Proceedings of CSCW '02*, NY: ACM Press, 21-30.
- 9 Handel, M., and Herbsleb, J.D. What is chat doing in the workplace? In *Proceedings of CSCW '02*, NY: ACM Press, 1-10.
- 10 Hudson, J.M., Christensen, J., Kellogg, W.A., and Erickson, T. Structure and Flow: “I’d be overwhelmed, but it’s just one more thing to do”: availability and interruption in research management. In *Proceedings of CHI '02*, NY: ACM Press, 97-104.
- 11 Isaacs, E., Walendowski, A., Whittaker, S., Schiano, D.J. and Kamm, C., The Character, Functions, and Styles of Instant Messaging in the Workplace. In *Proceedings of CSCW '02*, NY: ACM Press, 11-20.
- 12 McFarlane, D. Coordinating the interruption of people in human-computer interaction. In *Proceedings of INTERACT'99*. IOS Press, Inc., 295-303.
- 13 Nardi, B., Whittaker, S. and Bradner, E. Interaction and Outeraction: Instant Messaging in Action. In *Proceedings of CSCW '00*. NY: ACM Press, 79-88.
- 14 O’Conaill, B., and Frohlich, D. Timespace in the workplace: dealing with interruptions. In *Conference companion on Human factors in computing systems (CHI '95)*, NY: ACM Press, 262-263.
- 15 Schegloff, E. A., and Sacks, H. Opening and Closing. *Semiotica*, 8 (1973), 289-327.
- 16 Volda, A., Newstetter, W. C., and Mynatt, E. D. When conventions collide: the tensions of instant messaging attributed. In *Proceedings of CHI '02*, NY: ACM Press, 187-194.
- 17 Whittaker, S. Rethinking video as a technology for interpersonal communications: theory and design implications. *International Journal of Man-Machine Studies*, 42 (1995), 501-529.
- 18 Zhang, D., Lee, W. S. (2003). Question classification using support vector machines. In *Proceedings of SIGIR '03*, NY: ACM Press, 26-32.