Supporting Context-Based Chats for Enterprise Use

Tanuj Shah, Joyce Ohgi
Oracle Applications User Experience
[tanuj.s.shah, joyce.etsuko.ohgi]@oracle.com

Abstract

Personal chat has become immensely popular for personal use, but chat is less popular in the workplace. At work, chat is primarily used within teams and groups of people who already know each other. Personal chat is generally triggered from buddy lists between people who know each other. The current design and feature set of chat systems may be insufficient to support work-context based and enterprisewide communication and collaboration between members of an organization who do not know one another. This study analyzes a regularly appearing enterprise use case and proposes five design extensions to the traditional chat systems that would make these systems more useful in the enterprise environment. The Oracle Applications User Experience team's attempt is to facilitate enterprisewide collaboration between unknown members. We tested this new proposed chat designs using an HTML and JavaScript prototype with nine enterprise users and report the findings in this paper.

1. Introduction

Geographically dispersed teams can engage in virtual "corridor talk" by using a personal chat system [1,2]. Such chats are similar to the spontaneous, informal "corridor talk" that spring up in hallways or lunch areas [1, 2]. However, an important type of communication that must be supported in a enterprise environment is the one-time, spontaneous, work-related chat. This type of chat generally happens between two employees who may or may not know each other. Unlike personal chats that are primarily triggered from a buddy list [4], these spontaneous one-time, work-related chats are generally triggered by a specific context such as a work-related issue, document, or similar topic. In current situations users are likely to send an email to the other team members asking questions and requesting clarifications. However, this method results in an asynchronous, back and forth delayed information exchange. This process does not take advantage of the rapid and synchronous conversation aspect of a chat system [4].

2. Enterprise Use Case

After conducting user research that included interviews and observations, we identified a use case that occurs repeatedly in organizations. To explain this use case we describe one particular scenario. A manager views a technical document authored by a member of another team and wants to incorporate some of the document's knowledge into his or her own product. While reading the document the manager needs to clarify some information with the author of the document. Ideally, the manager needs his or her questions to be instantly clarified because the answers inform the manager's further understanding of the document and provide guidance to the next steps that the manager needs to take. The manager and the document's author work in the same organization but have never met each other and do not know each other formally. They have never communicated earlier. They work at different offices and in different time zones.

3. Extensions to Traditional Chat Systems

To support the previously described enterprise use case we propose the following five extensions to the traditional chat design that enables users to chat about one-time, spontaneous, work-related, and context-based conversations.

3.1. Embed Chat in the Context of Discussion

Chat conversations are generally initiated from buddy lists in traditional systems [4]. Traditional chat systems indirectly limit users to chat with colleagues that they closely work with because these systems require the authorization of both parties to begin a discussion. Further, buddy lists show the user's status to chat only to authorized members. Therefore, users require that the other member provide authorization to see their availability status [3]. This process creates a barrier between users who do not know each other but want to chat. Unlike personal chats, as described in the previous use case, enterprise chats are generally triggered when users request quick questions and clarifications from

CollabTech 2012 , August 27-29, 2012, Hokkaido, Japan.

colleagues in a work-related context[4]. For example, in the previously described use case, the manager wanted to obtain quick responses to questions about a work-related technical document. To remedy this problem, we propose adding an option as shown in Figure 1 to initiate a chat directly with the persons associated with the document. In our proposed design, the chat is initiated from the context itself rather than from a buddy list. This method omits the barrier of requiring users to authorize each other. Further, once the conversation between the chat users is complete, neither users are listed in each other's buddy lists.

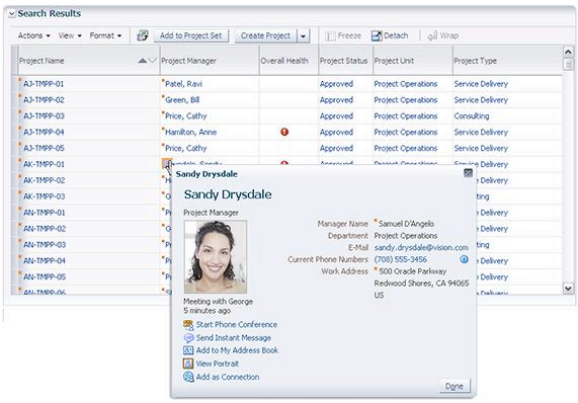

Figure 1: A contextual menu providing the ability to chat with a person listed in the document being viewed.

3.2. Use a Chat Title

One of the major challenges of chat-based systems has been to help users negotiate the availability of the person with whom they want to chat [3, 4]. The early messages in a conversation prior to discussing the main topic mostly concerns the availability to chat [4]. To support the negotiation of availability and because the chat could be initiated between members who may or not know one another, we propose adding a title to the chat. The title of the chat instantly indicates the context of the conversation and helps the chat receiver to decide whether to accept the conversation request, delay the response, or decline the conversation. We also propose adding a link to the context along with the chat title. This link provides the chat receiver with instant access to the document directly from the chat window. The chat title and web link to the document in conversation are automatically generated from the context in which it is being initiated. Chat initiators can further change the chat title to their liking. Figure 2 highlights the chat title and link generated by the system. Users can change the chat title at the beginning of the conversation or at any subsequent time during the chat, and this change would instantly be reflected in the chat windows of both the sender and receiver. While conducting user tests we found that the chat title helped chat receivers decide which chats to accept or reject. Many users reported that the chat title will also help them relocate the chat transcript for future reuse.

3.3. Ability to Respond Later

Users can determine the availability of other users to chat by noting the presence indicators in the chat window such as: Green for available, Yellow for not available and Red for do not disturb or away for a long time [3].

However, in spite of the presence indicators. if users receive a new chat at a unsuitable time, they can decline or delay responding and wait for a more suitable time to chat [4]. However, because the chat is context oriented and may be initiated between members who do not know each other, we propose an additional ability to convert synchronous chat into asynchronous chat. This provides users with the flexibility to determine whether to respond to the chat or continue to focus on their current work. During the chat conversation any side of the chat could use this Respond Later feature.

As shown in Figure 2, using this feature instantly sends a message to the intended chat responant stating that "User A has saved the conversation for later. Any new messages will be included when the saved chat is opened." The chat window instantly closes on User A's screen and the conversation becomes stacked into the saved conversations list. User A can now respond to the chat at a later time. User B in the meantime can continue to add messages to this conversation. User A can respond even if User B is offline. If User B is online, then User A's response is instantly seen on the screen, and if User B is offline, then User A's response is saved as a new message as a part of the saved conversations that appear after User B signs in. This feature extends chat functionality beyond negotiating availability. It encourages users to attempt for a synchronous conversation, but if that attempt fails it immediately shifts it to an asynchronous conversation similar to email.

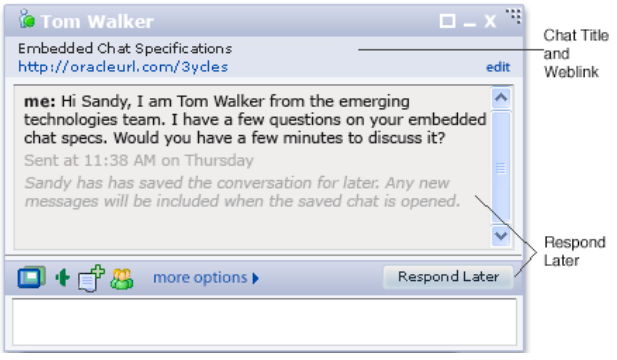

Figure 2: A chat window along with a chat title and a link to the context in discussion. The

Respond Later feature triggers an automated message in the chat window.

3.4. Screen Sharing

Because chats are based on a context in discussion it sometimes becomes important to share the screen and point out the particular item being discussed. A drawback of current chat systems is that they do not provide a visual of the workspace in conversation, and it sometimes becomes difficult to enter and explain everything [4]. For this, we added the ability to share screens. The drawback of other communication systems that allow screen sharing is that the conversations are not inititated from a context. The ability to have a shared view of the object in discussion helps users avoid misunderstandings and enhances clarity in chat conversations. The screen-sharing window by default starts from a small predefined size, and users can resize and reposition the window as required. This feature helps chat users to achieve common understanding quickly during a conversation by inserting screen shots directly into a chat window. Furthermore, some participants in the user tests mentioned that this feature would be helpful while conversing with the technical support teams when dealing with installation or other system issues. Figure 3 shows an example of the screen-sharing feature.

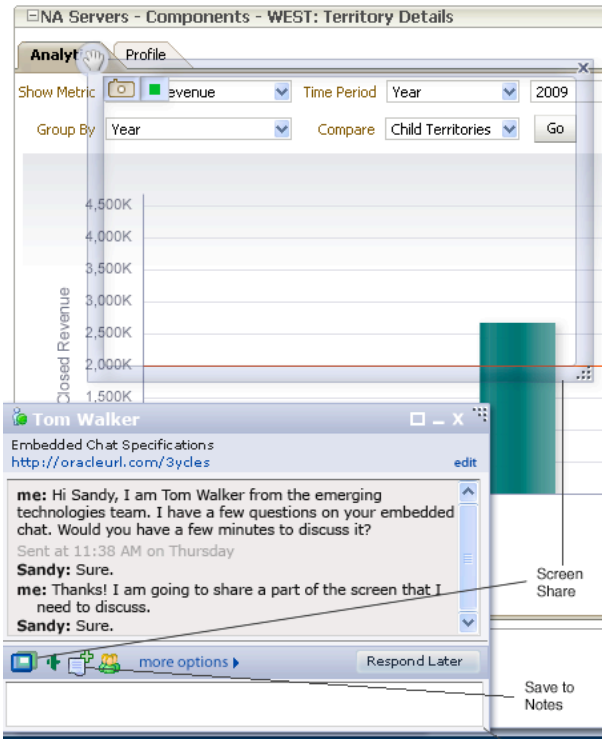

Figure 3: Screen Sharing and Add to Notes Feature

3.5. Save to Notes

Chat transcripts are likely to be used in the future as a reference tool because they contain valuable enterprise, business, or product information. Hence, it may be a good idea to support attaching the valuable part of the chat transcript to the context in discussion. For this, we propose a feature called Add to Notes that enables users to attach chat transcripts to the notes section of the context in discussion. The entire context-related chats in this section are listed by their chat titles. This feature extends the traditional "corridor conversation," which is a fleeting transaction by its nature [1, 2]. The Add to Notes feature helps to build a repository of useful information for reuse within the entire enterprise. Future users of this content can take advantage of the information derived from earlier chats regarding a particular topic.

4. The Embedded Chat Prototype

The Oracle Applications User Experience team developed an interactive HTML and JavaScript chat prototype called embedded chat. This prototype enables users to enter free-form text into a chat window and then the chat prototype would display a canned response into the window as if the response came from the user at the other end. This prototype was built for the above-specified enterprise use case. Using the prototype, users would request clarification on a piece of information from the author of the document in view. To gather the required information, users are asked to follow four steps. First, initiate a chat with the document author and check on the author's availability. Second, after negotiating availability users ask their specific questions for clarification. Third, users are required to share their screen shots pointing to the context of an issue or query the author with questions. to. Fourth, after resolving queries users must save the chat transcript in the notes section of the document using the Save to Notes feature. In addition, six similar tasks were created to test other features such as Repond Later, Editing the Chat Title, Repositioning the Screen-Sharing Window, and Sending a Screen Shot Using the Screen-Share Feature. These other tasks also helped users get accustomed to the entire new feature set.

5. User Test Findings

The Embedded Chat prototype was tested with nine participants. All nine participants fit the employee technical background user role, And all of the nine participants worked in organizations. In the preliminary questions we found that all of the users used chat on a

daily basis within their organization. However, they chat only with the coworkers they know. During the test session, participants attempted the seven tasks using the embedded chat prototype. All of the seven tasks were similar to the use case discussed in the previously mentioned Enterprise Use Case section. Seven tasks were formulated to test a different feature in each task and also to enable users to get accustomed to the new feature set. Overall, we found that embedding a chat in context, providing the ability to respond later, and making screen sharing available contributed the most to the participants liking. Users felt that these three features could lead to enterprise user adoption of chat. These users liked that the chat was integrated into acontext and all the required information and tools were now available on the same screen. They approved of the screen capture feature because it eliminated the steps of launching a new screen capture application and then going in to email to send the visual depiction of the problem. Furthermore, particiapants mentioned that in situations such as requiring technical assistance the screen share feature possibly performed better than a phone conversation. Four of the users responded that the screen capture feature would be handy for discussing troubleshooting use-cases. Screen sharing would be a powerful tool for supporting clarity and quality in knowledge transfer. The Respond Later feature provided the ability to think about a topic and respond later. In addition the Resond Later feature also worked as a reminder list to complete quick, small tasks at the end of the day. Two users mentioned that this feature made them feel like a chat is being converted into an email instantly as required. A few users also mentioned that though it created a lesser barrier in initiating the conversation, they would not continue to enter messages if the other user decided to respond later. We feel that it would take time for users to get accustomed to new feature. Furthermore, we found that it would take time for users to feel comfortable using all these new features. Finally, six of the users stated that with the possibility of such contextual chats they would chat with a coworker they've never met, even though none of them do that currently.

5. Conclusion

It is important to design enterprise chat systems that enable cross-enterprise collaboration over work-related contexts, leading to wide knowledge transfer and productivity at the same time. Apart from collaboration between members who know one another, it is equally important that collaboration is supported between members who do not know one another. Corporation can form policies to enable chat communications that are open across the organization. The chat feature embedded within context could show the availability status to anyone in the organization who has access to the contextual document pertaining to the other user. These systems may not require to authorize each other to initiate work-related discussions on chat. Further, chat conversations become more collaborative, interactive, and engaging because of the instant presence of context [4]. This leads to increased collaboration and knowledge sharing between users who normally would not communicate. Our embedded chat prototype received positive feedback from the nine user test participants and demonstrated the potential in allowing context-based conversations along with the powerful use of additional features. In future studies we hope to develop a working model of the proposed chat embedded into the work applications of an organization. Over time we could collect real data about its impact on enterprisewide collaborations, collaborations between unknown members, clarity in conversations with the use of the screen-sharing feature, and the Respond Later feature's ability to support the continuation of a conversation even if one of the user's has decided to respond later and the potential of the Save to Notes feature in creating an information repository for future reuse. Lastly, these proposed chat extentions are limited to a one-to-one chat and we hope to extend the chat design for use cases requiring group chat and multi-user collaborations.

9. Acknowledgement

The authors wish to thank John Cartan and Thao Nguyen of the Oracle Applications User Experience team for their help in developing these ideas.

10. References

[1] M.J. Handel, J.B Herbsleb, "What is Chat Doing in the Workplace", *CSCW '02 Proceedings of the 2002 ACM conference on Computer supported cooperative work*, ACM, New York, 2002, pp.1-10.

[2] D.G. Boyer, M. Cortes, M.J. Handel, "Presence Awareness Tools for Virtual Enterprises", *Proceedings of Object-Oriented Programming Systems, Language and Applications*, OOPSLA, Vancouver BC, 1998, pp. 1-6.

[3] J.D. Herbsleb, D.L. Atkins, D.G. Boyer, M. Handel, T.A. Finholt, "Introducing Instant Messaging and Chat in the Workplace", *In Proceedings of CHI 2002*, ACM Press, Minneapolis, 2002, pp. 171-178.

[4] B.A. Nardi, S. Whittaker, E. Bradner, "Interaction and Outeraction: Instant Messaging in Action", *CSCW '00 Proceedings of the 2000 ACM conference on Computer supported cooperative work*, ACM Press, New York, 2000, pp. 79-88.