

# Multimedia Chat for Helpdesks: a Requirements Study, a Practical SOA Architecture, and an Interface Prototype

Zon-Yin Shae<sup>1</sup>, Tony Bergstrom<sup>2</sup>, Claudio Pinhanez<sup>1</sup>, Mark Podlaseck<sup>1</sup>

<sup>1</sup>IBM T.J. Watson Research Center

<sup>2</sup>Department of Computer Science, University of Illinois at Urbana-Champaign  
{zshae, pinhanez, mark}@us.ibm.com, abergst2@cs.uiuc.edu

## Abstract

*In this work we study the actual requirements of a multimedia chat system for helpdesks by quantifying the verbal imagery communication in 115 real transcripts. The data shows that in chats which use verbal imagery, it corresponds to about 25% of the total time, most of it (75%) coming from the agent. These results clearly show that multimedia support for contact center chat systems is called for. But unlike in personal chat systems, the existing 3<sup>rd</sup> party chat services for helpdesk centers are text-based only. To circumvent this issue, an innovative, practical SOA-based chat system architecture is proposed in this paper that creates multimedia chat services by decomposition and re-composition of the services from an existing 3<sup>rd</sup> party text-based helpdesk chat product. Our SOA architecture implements uniquely a session-based interactive mashup mechanism and bundles it to the previous operations in the same chat session. However, to be effective in the context of technical support, a multimedia chat system must be provided with user and agent interfaces which address the distinctive aspects of technical support communication and imagery use in the conversations. We present an interface prototype that address those issues through the use of persistent imagery, linking instructions and imagery, use of common images, and easy screen sharing.*

## 1. Introduction

Today's contact centers are dominated by phone calls as the primary channel through which end users reach the help desk. To survive in a competitive world, organizations that run contact centers are looking at alternative communication channels and mechanisms to improve efficiency and reduce costs relative to the cost of servicing phone calls. Chat has become increasingly important as an effective mean to connect to contact centers. However, the buddy-list chat architecture used in personal instant messaging systems (such as *Lotus Sametime*, *Yahoo! Messenger*, *AOL Instant Messenger (AIM)*, or *Microsoft Messenger (MSN)*) is not applicable to the contact center

chat systems since it requires, among other things, mechanisms for routing chat requests to agents.

In this paper we initially discuss the need for multimedia in contact center scenarios, in the context of technical support, by first examining the literature on visual communication. Specifically, text-only technical chat relies on both the user and the agent being able to communicate in a common language to reference physical and on screen objects. However, the user can have a vocabulary that differs extensively from the often highly technical vocabulary of the agent. In some circumstances, the user might not even have words to describe his or her current state.

*Common ground* [2][3] provides shared understanding of references, allowing us to refer to a chair and not have to describe what a chair is. However, technical support starts with an unbalanced knowledge by default: expert (agent) and non-expert (user). An agent must be able to understand the user's problem, described in terms that are familiar to the user. Similarly, the user must be able to understand questions and instructions from the agent, who has intimate knowledge of what should be happening, provided the user follows instructions exactly. Text-only chat technical support would take a much longer time for agent and user to reach their common ground.

Our work seeks a ways to provide a lightweight verbal imagery coordination of technical support to enable the illustration of communication in establishing common ground much faster between user and agent. In this paper, we present an analysis of 115 real technical support chat text logs that establishes clearly the need and usefulness of imagery in support of technical chat services.

We then look into how to introduce multimedia into text-only chat systems which comprise most of the existing 3<sup>rd</sup> party chat products for contact centers. It would be a very expensive approach to totally rewrite the code to convert the current products for supporting multimedia in chat services.

To overcome this issue, we present an innovative SOA (Services Open Architecture) approach for providing multimedia chat services by reusing existing text-based only products. We analyze the entire services components of a typical chat service product and decompose the chat system into 4 major components: user facing GUI component, agent facing GUI component, session routing component, and media routing component. This services decomposition enables the establishment of a mechanism for a services re-composition process and therefore lays a solid foundation for devising a SOA-based multimedia chat architecture.

Web-page mashup mechanisms are currently very popular in Web 2.0 and in synergy with the services composition mechanism in SOA. Web-page mashup is becoming a very powerful and effective approach to create new web applications out of the content of existed web pages. A regular web mashup is memory-less in the sense that each operation is independent of the previous one. We implement a session-based interactive mashup for the user GUI component in our chat SOA architecture in which the current mashup operation is related and bundled to the previous operations in the same chat session. This session-based interactive mashup makes our SOA chat mashup solution unique and especially interesting.

However, providing a fully multimedia technical support via the Internet constrains interaction by the need to be universally accessible. Interfaces are often lightweight and attempt to be accessible by the most common browsers without requiring numerous changes in security policies for processing multimedia content. As an added technical constraint, call centers are often located overseas from the client. This creates an issue with bandwidth and delay. Streaming video can be impractical due to bandwidth, disallowing or discouraging remote views of desktops; while delay hinders the communication strategies of the coordinating user and agent [4]. Shared screens, or remote viewpoints, provide information needed to establish what the user is actually seeing; however, due to the complexity of this task for many users, a fully multimedia augmented help is often reserved for only the most complex problems.

While our SOA multimedia framework would be able to support fully multimedia augmented communication, we focus in this paper investigating the effectiveness of verbal imagery technology to optimize the cost and provide the best benefit of the multimedia chat system to users. With that in mind, we present and discuss the prototype design of a image-augmented technical support chat system, looking into issues in the design and development of generic imagery augmented chat systems. We then discuss the characteristics of illustration to aid and augment text-based conversation.

The paper is organized as follows. Section 2 outlines the visual reference communication; Section 3 describes the chat verbal imagery analysis; Section 4 discusses the SOA architecture for services composition into a new chat services system. Section 5 describes the interactive mashup implementation. Section 6 decomposes the chat system into services components and Section 7 discusses a prototype design for studying the verbal imagery chat. Section 8 concludes the paper.

## 2. Visual Reference Communication

Imagery provides visual references to aid communication, tell stories, demonstrate ideas, give directions, educate, warn, signal and illustrate. Most direct human communication relies on visual elements in addition to verbalization [1]. However, as new technologies such as telephones, email, chat, and text messaging developed, imagery increasingly is seconding verbal language.

Web cameras and media spaces [5] allow people to see each other from remote locations. They establish a visual connection but encounter hardships of creating a full connection between people due to eye gaze, camera position, limited view angles, etc. Continuing work in the field has examined the idea of *common ground* and shared workspaces has been covered extensively in remote collaboration and CSCW literature [4][6][7][8]. These efforts have relied on sharing a view space and allowing participants to interact verbally while providing capabilities for deictic references. In most cases, the imagery provided is a constant video stream of the workplace and other setups that are not feasible for most technical support centers or clients.

Work by Voids, et. al. examined how people use snapshots to illustrate, communicate, and punctuate their conversations [9]. They provided an interface that directly embedded a web cam into text chat; snap shots are inserted inline with the conversation to provide posed images. In this manner, people could use their cameras to provide custom emoticons, visual cues, and tell stories. The still photos proved more meaningful and expressive than a continuous web cam video.

Online photo sharing sites and social network sites that support photo-sharing allow people to show, share, and discuss their pictures with others. Additionally, some offer tools to reference directly into the images [10]; allowing people to point and direct other viewer's attention without a long verbal description. Other image based websites use the images and relative position on shared public pages, to directly interact and create visual dialogs [11].

Imagery can be injected into conversation as conversation directly, as in the case of "emoticons", or as part of a

graphical notation system. Recently, Chinese artist Xu Bing developed a pictorial language in his work *A Book from the Ground* [12]. Based on info graphic icons from airplane emergency procedures and warning materials, this visual language creates a linear text that can be read independent of spoken language. Bing has incorporated this language into a customized chat client. Finally, work by Davis studied how to depict actions verbs through icons for annotation of video [13].

Many of the most popular personal chat systems, such as *Lotus Sametime*, *Yahoo! Messenger*, *AOL Instant Messenger (AIM)*, or *Microsoft Messenger (MSN)* allows the inclusion of imagery in the conversation, in different forms, including emoticons, desktop snapshots, picture and photo pasting, or free drawing. Lack of conversational imagery is characteristic of remote chat and phone-based technical support systems. In this multi-billion dollar industry striving for higher degrees of efficiency and agent utilization, the lack of initiatives, or even prototypes, for imagery augmented chat systems is surprising.

### 3. Chat Verbal Imagery Analysis

Here, we first quantify the need of imagery in this context. To study this issue, we randomly selected 115 technical support sessions from a set of 1244 logged text chat conversations between agents and users. Our corpus represents support sessions from December 5, 2006 to January 23, 2007. The conversations refer to IT technical support provided by a helpdesk service for all employees of a large size Fortune 500-company. This help desk also performs other forms of technical support, such as phone support. Chat support was offered for limited topics pertaining to general issues; clients (users) accessed chat support via a web based form, offering basic messaging capabilities.

We manually coded 115 sessions for:

*Session Duration*: Measured from the time the agent initiates conversation about the client's problem to the time of the final message.

*Verbal Imagery*: Reading through our corpus, we noted the use of language to describe visual features. This includes descriptions of menus, locations of icons, and general descriptions of what is seen on the screen. Each instance of verbal imagery was noted along with the following 3 details (a, b, and c):

(a) *Source*: We define the source to be the individual describing a visual reference. If the agent were to prompt the client to *describe what you see*, the client would be the source, though the agent initialized the instance of *Verbal Imagery*.

(b) *Duration*: We used a conservative measure for duration. An instance of verbal imagery begins with a visually descriptive comment or with a prompt to visually describe what is seen. The end of an instance is either acknowledgement that the imagery was understood or that conversation moves on. We measured time based on the timestamps of these two comments. In most cases, acknowledgment will come in the form of "ok" or "did that." In some cases, the agent will press on with the next step, assuming the client has not had issues with the previous instruction,

We time to the beginning of that comment. In other cases, the client makes a comment that does not acknowledge completion but continues the interaction; these cases are less common and we only time to that comment.

(c) *Confusion*: We noted verbal imagery in which either the agent or the client indicated confusion or misunderstanding. We were looking for instances when it was apparent that the visual description was not effective. Generally the client would ask for further clarification. This included comments such as "I don't see what you're talking about" and "How do I do that."

Additionally, as the agents in our corpus could request a passive remote viewpoint via internal software or *Netmeeting*, we resample the corpus and selected 18 instances from the 55 elements attempting remote viewpoints. We coded these sessions to determine some base level characteristics: duration and the time to set up remote connection (measured from the initial view request to the first comment that established connection has been made). This data was used to give an impression of how effective this form of imagery is in text-based technical support.

We divide sessions into two categories, those with *Verbal Imagery* (60) and those with *No Imagery* (55). A striking difference in session length becomes apparent with a t-test [14]. The use of *Verbal Imagery* correlates with an increased session length of 24.9 minutes while *No Imagery* is less than half of that length at 11.8 minutes ( $t(113)=3.57$ ,  $p<.001$ ). The t-distribution with 113 degrees of freedom indicates the probability the two means are different by chance is less than 0.001. Looking only at *Visual Imagery* conversations, we found that 6.25 minutes (25%) were spent per session describing some kind of visual. A plot of all sessions can be seen in Figure 1 which shows that the amount of verbal imagery varies greatly in each chat session: ranging from a high of roughly 60% to a low around 1%. The 55 non-imagery sessions plotted along the axis, visualize the shorter average session length for those conversations.

Agents favored using verbal imagery over the clients: 75% of all time spent using words to convey imagery comes

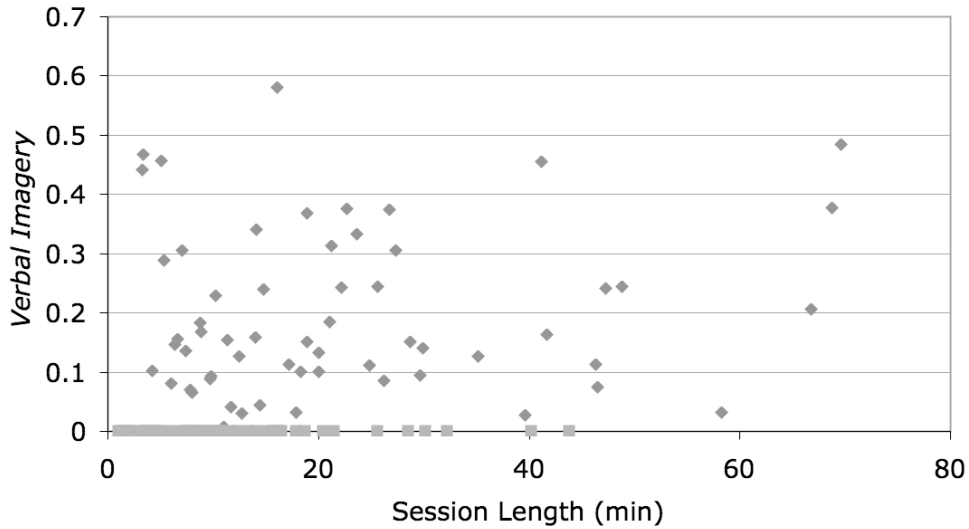


Figure 1: Verbal imagery session length distribution.

from the agent. Often this imagery describes what the client should be looking for on their screen. When that imagery creates confusion, the time spent discussing that imagery to create a common ground increases: instances with confusion average 130 seconds ( $\sigma=129$ ), compared to the 74 second ( $\sigma=88.9$ ) average over all instances of imagery. Below is a 245 second example of this kind of conversation, in reference to a commonly used program in the company.

[11:13:54] **Agent:** Lets click on Start now  
 [11:15:15] **Client:** Ok  
 [11:17:01] **Agent:** Do you get the same error even now?  
 [11:17:22] **Client:** Where is start?  
 [11:17:59] **Agent:** On the replication Page

Session length average increases from 24.9 minutes to 51.4 minutes ( $\sigma=23.4$ ) when conversations utilize desktop sharing. We do not know for sure if these problems were more difficult or if there was more confusion as we did not code for anything but time in these 18 samples. However, we should also note that the average time just to set up this connection using pre-installed company standard software is 9.5 minutes ( $\sigma=5.5$ ), nearly the length of a full conversation without imagery.

The findings above clearly justify the need of imagery in technical support based on chat systems. However, as commented before, few of the existent systems for contact center do not provide functionality for that. In the next section, we present a method to bypass the limitation.

#### 4. Decomposing Chat services Components

Most of the existing text-based chat products for contact centers are not SOA based systems. At the first grasp, there seems to be no reusable component and no other way than to totally re-write the code to support the multimedia. This totally code re-write approach presents an unsolvable challenge since these 3<sup>rd</sup> party product source code is not available.

Alternatively, creating a brand new multimedia chat system product from scratch also presents a big investment and long delay to market. We solve this problem by carefully analyzing and decomposing the existing systems into 4 services components (shown in Figure 2): user facing GUI component, agent facing GUI component, session routing component, and media routing component. This decomposition mechanism enables and lays the solid ground for easily SOA services re-composition and integration.

The *session routing* component is used to allocate a free agent whose skill set is the best to solve user's problem in the session, then route the user's session request to the allocated agent. The *media routing* component is initiated between user and agent immediately after the session routing component completed. This component is used for routing the media content between user and agent. The *user facing GUI* component provides the GUI interface for user, and the *agent facing GUI* component provides the GUI interface for agent.

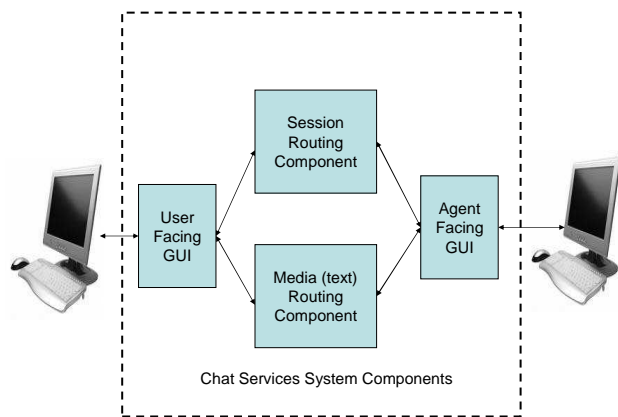


Figure 2: Decompose chat services components.

## 5. SOA Chat Services Composition

The new multimedia chat services (shown in Figure 3) is a SOA services composition reusing the existed 3<sup>rd</sup> party chat product (serves as new session routing component) and the other 3 new services components: the *interactive mashup* component (new user facing GUI), the *multimedia routing* component (new media routing component), and the *multimedia agent facing* component (new mashup agent facing GUI).

The SOA chat services system composition consists of the architecturally 4 identical components (user facing GUI, agent facing GUI, session routing component, and media routing component), but with the entire original 3<sup>rd</sup> party chat product (in Figure 2) nested in the new composition services (in Figure 3) to provide the services of a session routing component. This architecturally identical property between decomposition (Figure 2) and composition (Figure 3) is very important in our design architecture which is enabled by chat services component decomposition mechanism. It indicates this SOA chat composition architecture can be nicely scaleable to more levels of nested compositions.

Our unique SOA services composition architecture mechanism separates the session control path (session routing path) and the media path to reuse the chat session routing engine of existing 3<sup>rd</sup> party chat product. Separation of control path and media path can provide efficient and low cost multi-language and multimedia support. This unique SOA architecture can enable the integration of customized enterprise back-end applications, for example, ticketing systems integration (which automatically creates a ticket when chat start and store the entire chat transcript into the ticket when chat end), or of knowledge content databases.

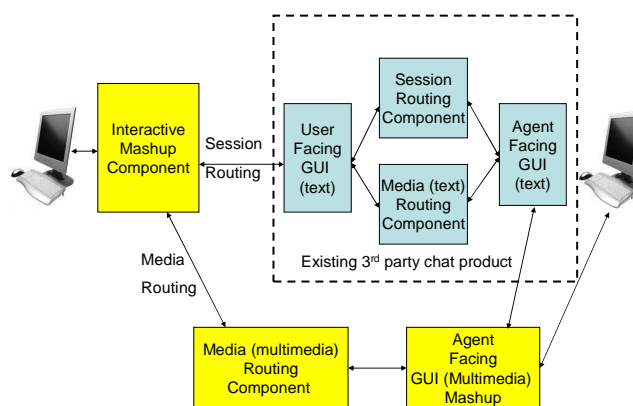


Figure 3: SOA chat services system composition.

## 6. Session-Based Interactive Mashup

Both user facing GUI mashup and agent facing GUI mashup components have the similar interactive mashup property, therefore, the discussion of session-based interactive mashup in this section apply to both components. The interactive mashup component (in Figure 3) is a user facing GUI component which is a mashup content from the user facing GUI in the nested 3<sup>rd</sup> party chat product. The user facing GUI of the nested 3<sup>rd</sup> party chat product is basically a web page with form submission to carry the user chat input content to the chat system, and also display the response from the agent to user. To support a bidirectional interactive mashup for chat, the mashup needs to support awareness of “chat session”, that is, it needs to keep track of the related chat session information for each HTTP interaction.

When the interactive mashup component receives a web page response from the user facing GUI, it parses the HTML content of the web page and extracts the real content from agent. It also will extract the chat session information, for example, chat session ID, from the received web page. Then it mashes the extracted agent content and the chat session information into a new web page of its own and presents it to the user. The session information is encoded in the new web page to the user in such a way that when the user submit a new HTTP request, these chat session information can be sent back to the interactive mashup component. Consequently, when user submit a new chat content by sending a new HTTP request to the interactive mash component, the chat session information and the actual content from user is extracted, and a new HTTP request is composed and sent to the user facing GUI. As such, the interactive mashup component can keep track and communicate all the chat session information between it and the user facing GUI.

The session-based interactive mashup makes our SOA chat mashup unique in the sense that it can use multiple 3<sup>rd</sup> party chat products at the same time, albeit in different sessions. In the session-based SOA chat services composition (shown in Figure 4) where there are 4 existing 3<sup>rd</sup> party chat products from which one can choose the services for composition. The session-based SOA composition requires the services composition to be done only at the session level, and not in the individual mashup request level. For example, if the first chat interactive request of chat session 1 choose 3<sup>rd</sup> party product A for service, then all the subsequent chat interactive requests belong to session 1 should always choose 3<sup>rd</sup> party product A. The same rule applies to chat session 2 which chooses, for instance, to start with 3<sup>rd</sup> party product D. Note that the individual chat requests from session 1 and session 2 arrived interleaving at the interactive mashup component.

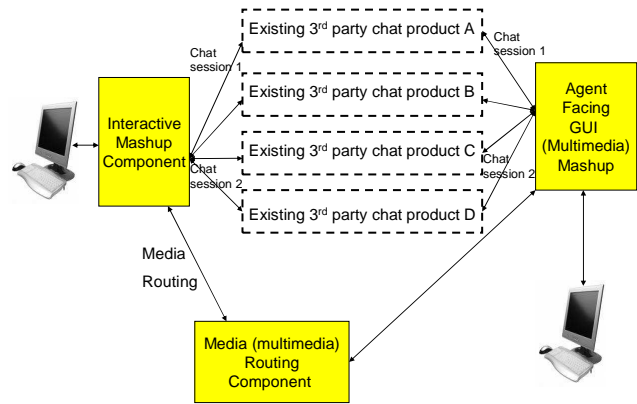


Figure 4: Session based SOA chat services.

## 7. Prototype Design

Having established the need of imagery in technical support chat, and engineered a way to inject multimedia into existent 3<sup>rd</sup> party products for chat, we focused now on designing the experience of using imagery in a technical support chat.

From the initial idea of replacing cumbersome verbal imagery with actual imagery, we created prototype for the user and agent facing GUIs to explore the experience. Our initial design incorporated characteristics of a standard chat interface with the imagery augmentation.

As with most chat clients, our system allows individuals to compose and evaluate their messages before sending them. A message in this case is not only text but the image selected and the location of any spatial notations. Before a message is sent, one can change images, move and resize the spatial note, or examine past imagery.

Figure 5 shows a screen shot of our prototype. (A) This is the chat window, which differs from traditional chat windows by allowing the user to mouse over any item to reveal the associated image on the B window, as well as any spatial references shown on the image. Items in grey indicate a box has been drawn on a part of the image for that comment. (B) The main image space shows the latest images and highlights the last comment by showing its text. Any comment can be examined by mousing over the box. (C) This box provides images that are meaningful to the problem at hand, allowing either the agent or client to quickly select an illustration.

Images come from three sources: screen grabs, previously used imagery, or the common image bar (Figure 5C). Using the “Screen Grab” button at the bottom of window A,

screen grabs are captured and can be manipulated in the visual space or enhanced with notes or references (Figure 5B) before sent to the dialoguer. Previously-used imagery can be utilized by making another note or just sending it with the next message. Lastly, the common image bar provides a set of imagery that is considered appropriate for the current problem. One can simply click on the images to bring it into the visual space.

Once imagery is sent to either client or agent, the new image is displayed immediately with any spatial notes highlighted as in Figure 5B. Past imagery and spatial notes can be revisited and reviewed in a few ways: one can mouse over other spatial notes on the current image to display the text; one can mouse over text marked with a grey background (Figure 5A) to display the imagery and notes associated with that message; or one can select an image from the common image bar and see what spatial notes exist.

Imagery in our prototype is both separate and integrated into the conversation. We separate to avoid inline imagery, as inline images interrupt conversation flow and push recent history out of the current window. However, mouse-overs allow to easily integrating directly the text chat into the imagery.

As seen in the Section 2 there is a significant effect of verbal imagery. Our prototype design seeks to support a lightweight illustrated chat mode, in which one can converse in text but visually highlight thoughts and illustrate problems when needed. Here we discuss four prominent aspects of the design.

*Persistent imagery* offers potential to discuss and reference the image. Displaying the image in a frame that does not advance with the text allows the individuals to continually view and reference the image as conversation progresses.

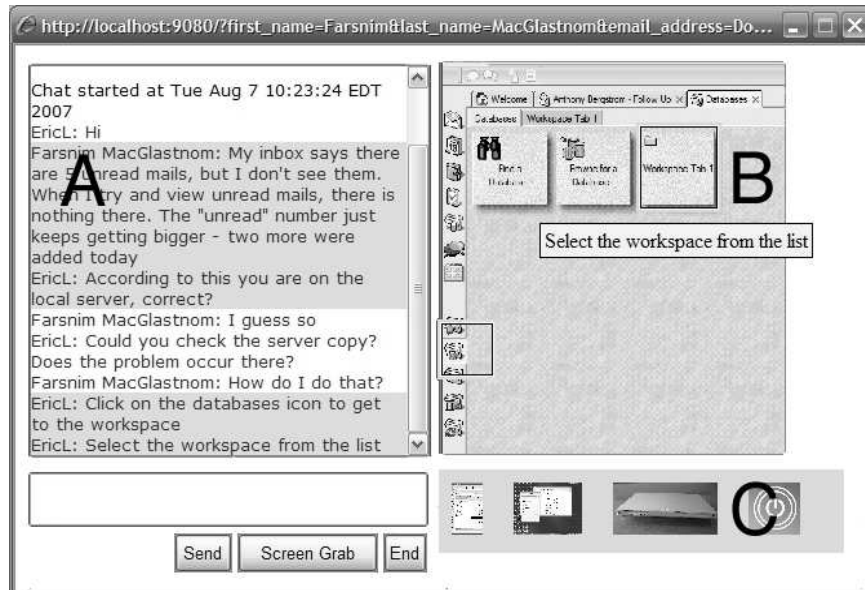


Figure 5: Screen shot of prototype of user GUI chat augmented with verbal imagery.

Additionally, by not demonstrating the image inline, users and agents can continue their conversation with full view of their conversational context.

*Linking instructions and imagery* supports interaction that is akin to deictic referencing. By highlighting a section of the image and tagging it with a message, participants can point and direct attention appropriately. Instead of direct references to refer to objects, the chat corpus we coded would reveal indirect descriptions:

*“Right Click on the Red/Green icon (Icon that has the letter “i” with white background) that you have on the right bottom corner of your Desktop.”*

With shared imagery in mind, one could show the image with the message *“right click this icon.”*

*Common images* reference the fact that many technical support conversations are repetitive: multiple people encounter similar problems, and the solutions to different problem can follow the same initial path to fix. We provide a common image collection to store images most relevant to the task at hand. At present we populate this bar in the interface with pre-selected images for a scenario; in future iterations this can be scoped by the agent, or automatically by a image search software able provide to find appropriate imagery based on the topic of the session or the words being typed.

*Easy, privacy-preserving screen sharing* provides quick, easy to access, reference points. Our prototype supports a basic mechanism to capture screen image and zoom in on relevant sections. Prior to being sent to the other party, one can examine and decide if the image is appropriately illustrative: one can also retain privacy by reviewing the

screen capture to ensure private data is not shown. Planned but not yet implemented extensions would allow cropping and blurring of sensitive data to enhance privacy features.

We believe that our GUI prototype, even in its current primitive stage, emphasizes four criteria for success in delivering imagery as support communication.

*Predefined scripts and prearranged imagery:* as 75% of all imagery was initiated by the agent, predefined text and imagery could be set up or preconfigured to display in each session. For example, as the agent often requests clients using Windows systems to launch programs by selecting “Run” from the “Start” menu and typing the program name into the box: this message can be paired with appropriate imagery. In examples that require navigation of menus and preferences, some phone support desks already utilize imagery for the agent, so that the agent can see what he or she needs to describe the client. Sharing similar imagery with the client can lead to a more thorough understanding with less verbal translation by the agent.

*Illustrations to reduce confusion:* people don’t want to appear stupid, confused, or incompetent. As a result, rather than asking for clarification people become frustrated when unable to answer a question or find the reference of the remote agent. With appropriate illustrations accompanying every verbal exchange, a basic common ground is provided to clarify without having to explicitly request more information.

*Privacy management:* Sharing a remote view of a personal desktop with another person raises issues of privacy. With many windows open or other programs running in the

background, there can always be a concern that some confidential material might appear in an email, instant message, or just a file that one forgot to close. Allowing the user to preview the images before sending it allows one to filter images that show material that they should not.

*Lightness*: A system like our prototype is lightweight and transparent. The client can see anything that is shared, and there is never a question, like those that can arise in remote administration, of “*did that agent open up a back door while he had access to my system?*” As a side effect of the system, the client must then do all actions on their own, allowing clients to learn how to deal with their problem. A future angle on this approach could use this method to train or teach at a distance.

## 8. Final Remarks

This paper studies the problem of improving call center chat systems by the use of multimedia and presents an almost complete analysis of the process. We start by quantifying the cost of lack of imagery handling in technical support chat systems, then propose an architecture to create such systems on top of existing 3<sup>rd</sup> party chat systems, and finalize by presenting a prototype for the interaction experience. Although our work lacks usability testing and actual performance tests, we believe that it is quite rare to see works in this area with similar breadth.

To understand the benefit of the multimedia chat system for helpdesk, we analyze verbal imagery context of chat sessions to demonstrate the need for real imagery in technical support chat. To the best of our knowledge, this is the first time that a verbal imagery requirement has been quantified by chat text analysis.

An innovative SOA chat services architecture for help desk has been proposed that can create multimedia chat services by the services composition of the existing 3<sup>rd</sup> party text-only chat products. By understanding and analyzing the existing chat products for helpdesk, we devise a chat services decomposition and re-composition architecture which employs of a novel session-based mashup scheme. However, the chat SOA architecture presented in this paper does not to reuse the media routing component. It is obvious that the text-only routing component cannot be reused for the multimedia routing component due to the lack of standards for end to end media streaming mechanism, storage, and protocol, but is currently under our further study.

We also show in our prototype that the use of imagery in the GUIs of technical support require a careful analysis of the conversational aspects of the interaction. We identified four criteria for success in delivering imagery to enhance technical support: predefined scripts and imagery; use of

illustrations to reduce confusion; privacy management; and lightness. Based on these criteria, we centered our design on four principles: persistent imagery; linking instructions and imagery; use of common images; and easy screen sharing. Further development of the user’s and agent’s GUIs, testing, and deployment are in plan.

As we demonstrated, enormous potential exists in using imagery in technical support. We are focusing our future work on improving the user experience in multimedia-augmented chats, as well as on the underlying mechanisms to enhance the agents’ productivity.

## 9. References

- [1] Mehrabian, A.: *Silent Messages*. Wadsworth (1971)
- [2] Clark, H.H.: Grounding in communication. In Resnick, L.B., Levine, R.M., Teasley, S.D., eds.: *Perspectives on socially shared cognition*. (1991) 127–149
- [3] Clark, H.H.: Definite reference and mutual knowledge. In Joshi, A.K., Webber, B., Sag, I., eds.: *Elements of discourse understanding*. Cambridge University Press (1981)
- [4] Fussell, S.R., Kraut, R.E., Siegel, J.: Coordination of communication: effects of shared visual context on collaborative work. In: *Proceedings of CSCW*, New York, NY, USA, ACM Press (2000) 21–30
- [5] Bly, S.A., Harrison, S.R., Irwin, S.: Media spaces: bringing people together in a video, audio, and computing environment. *Commun. ACM* **36**(1) (1993) 28–46
- [6] Ishii, H., Kobayashi, M.: Clearboard: a seamless medium for shared drawing and conversation with eye contact. In: *Proceedings of CHI*, New York, NY, USA, ACM Press (1992) 525–532
- [7] Kirk, D., Rodden, T., Fraser, D.S.: Turn it this way: grounding collaborative action with remote gestures. In: *Proceedings of CHI*, New York, NY, USA, ACM Press (2007) 1039–1048
- [8] Kraut, R.E., Gergle, D., Fussell, S.R.: The use of visual information in shared visual spaces: informing the development of virtual co-presence. In: *Proceedings of CSCW*, New York, NY, USA, ACM Press (2002) 31–40
- [9] Vaida, A., Newstetter, W.C., Mynatt, E.D.: When conventions collide: the tensions of instant messaging attributed. In: *Proceedings of CHI*, New York, NY, USA, ACM Press (2002) 187–194
- [10] Flickr: (<http://www.flickr.com>)
- [11] McDonald, D.W.: Visual conversation styles in web communities. In: *Proceedings of HICSS*, Washington, DC, USA, IEEE Computer Society (2007) 76
- [12] Bing, X.: *Book from the ground*. In: *Automatic Update*. Museum of Modern Art (2007)
- [13] Davis, M.: *Media Streams: Representing Video for Retrieval and Repurposing*. PhD thesis, Massachusetts Institute of Technology (1995)
- [14] Weiss, N. A., *Introductory Statistics*. 4<sup>th</sup> ed. Addison Wesley Publishing Company. 1997.