In 2008 IEEE International Conference on Services Computing (SCC 2008)
July 8-11, 2008, Honolulu, Hawaii, USA

Multimedia Chat for Helpdesks: a Practical SOA Architecture

Zon-Yin Shae[1], Tony Bergstrom[2], Claudio Pinhanez[1], Mark Podlaseck[1]
[1]*IBM T.J. Watson Research Center*
[2]*Department of Computer Science, University of Illinois at Urbana-Champaign*
{zshae, Claudio, mark}@us.ibm.com, {abergst2}@cs.uiuc.edu

Abstract

The existing 3rd party chat services for help desk are text based only and more effective communication between user and agent for better help desk support in multimedia chat services is called for. It would be a very expensive approach to totally re-write the code to convert the current text based chat product for supporting multimedia in chat services. An innovative SOA chat system architecture is therefore presented in this paper to create a new multimedia chat services for help desk by novel decomposition and re-composition of the services from the existing 3rd party text based chat product. A regular web mashup is memory-less which the current mashup operation is independent from the previous one. Our SOA architecture implements uniquely the session based interactive mashup mechanism which the current mashup operation is related and bundled to the previous operations in the same chat session.

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 [1]. 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.

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.

The existing 3rd party chat products for help desk are text based only. It would be a very expensive approach to totally re-write the code to convert the current product for supporting multimedia in chat services. In this paper, we present an innovative SOA (Services Open Architecture) approach for providing multimedia chat services by reusing existing text based only products without changing their codes. We analyze the entire services components of a chat help desk services by nobly decomposing the chat system into 4 major services components: user facing GUI component, agent facing GUI component, session routing component, and media routing component. This uniquely services decomposition enables the mechanism for the services re-composition process, and therefore lays a solid foundation for devising SOA based multimedia chat architecture.

Web page mashup mechanism is 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 which current mashup operation is independent of the previous one. We implement a session based interactive mashup for user GUI component in our chat SOA architecture which 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 unique and especially interesting.

The paper is organized as follows. Section 2 decomposes the chat system into services components; Session 3 discusses the SOA architecture for services composition into a new chat services system. Section 4 describes the interactive mashup implementation. Section 5 concludes the paper.

2. Decompose 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 3rd 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 1): 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.

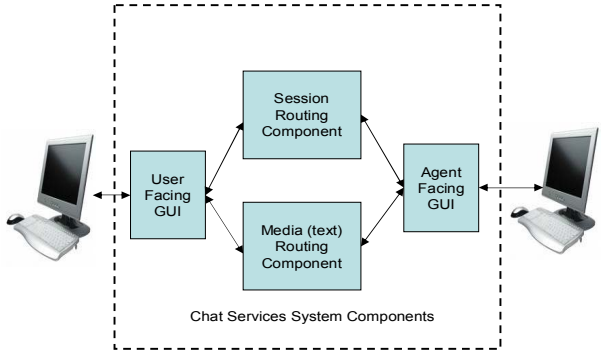

Figure 1: Decompose chat services components.

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.

3. SOA Chat Services Composition

The new multimedia chat services (shown in Figure 3) is a SOA services composition reusing the existed 3rd 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 3rd party chat product (in Figure 1) nested in the new composition services (in Figure 2) to provide the services of a session routing component. This architecturally identical property between decomposition (Figure 1) and composition (Figure 2) 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.

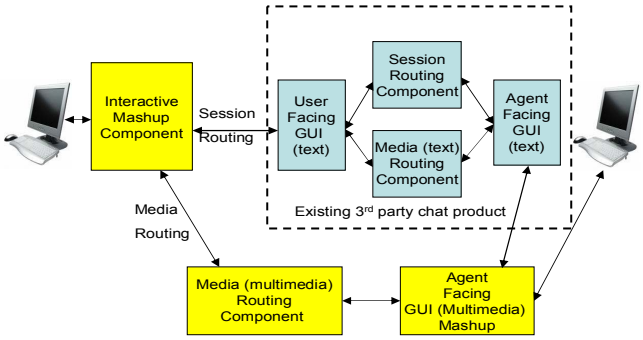

Figure 2: SOA Chat Services System Composition

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 3rd 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.

4. 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 3rd party chat product. The user facing GUI of the nested 3rd 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.

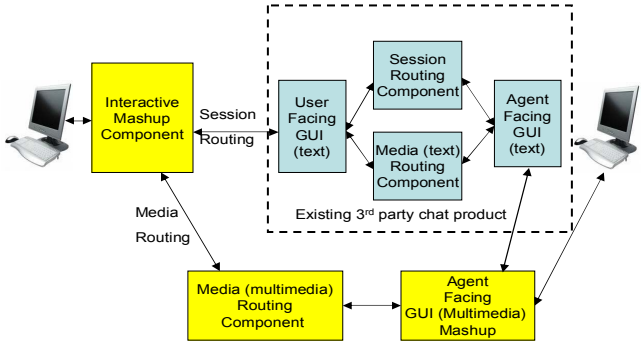

Figure 3: SOA chat services system composition

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 3rd 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 3rd 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 3rd party product A for service, then all the subsequent chat interactive requests belong to session 1 should always choose 3rd party product A. The same rule applies to chat session 2 which chooses, for instance, to start with 3rd party product D. Note that the individual chat requests from session 1 and session 2 arrived interleaving at the interactive mashup component.

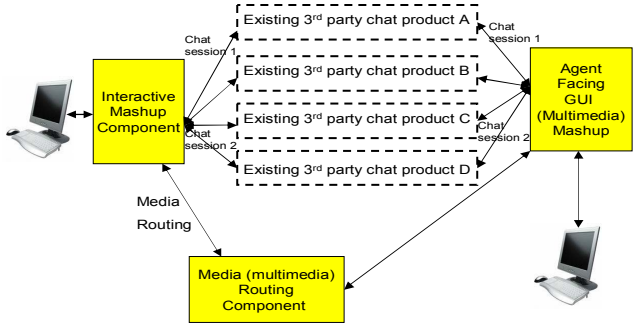

Figure 4: Session based SOA Chat Services

5. Final Remarks

An innovative SOA chat services architecture for help desk has been presented that can create a new multimedia chat services by the services composition of the existing 3rd party text only chat products. By understanding and analyzing the existing chat products for help desk, we devise chat services decomposition and re-composition architecture which consists of unique session based mashup scheme to make this possible. The chat SOA architecture presented in this paper assumes not to reuse the media routing component. It is obvious that the text based only routing component can not be reuse for the multimedia routing component. Reusing multimedia routing component is currently under our further study due to the lack of standards for end to end media streaming mechanism, storage, and protocol.

6. References

[1] Shae, Z., Garg, D., Bhose, R., Mukherjee, R., Guven, S., Pingali, G.,: Efficient Internet Chat Services for Help Desk Agents. SCC 2007, Salt Lake, USA (2007) 589–596
[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 Universit Press (1981)