Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 475–479
ISBN 978-83-60810-22-4
ISSN 1896-7094

Recording/Archiving in IBM Lotus Sametime based Collaborative Environment

Sunil John, Sorin Ciolofan and Cheryl Anne Harding
IBM Dublin Software Lab
IBM Technology Campus
Dublin, Ireland
Email: suniljoh,ciolofan,hardinga@ie.ibm.com

Peter Westerink
IBM Watson Research Labs
Hawthorne, New York
United States
Email: peterw@us.ibm.com

***Abstract*—Archiving video conferences is an important feature that allows storing conferences for later playback or for any other possible usage. This part of videoconferencing has never been an active area for most conferencing solutions. Many aspects of archiving conferencing solutions, including application sharing as well as document sharing among the participants, are not properly considered. In this paper we discuss IBM Streaming Engine based archiving solution. This recording and archiving facility leverages IBM Sametime collaborative conferencing solution. Apart from the discussion on the archiving facilities, a brief description of IBM Streaming Engine as well as its features is presented.**

***Index Terms*—Collaborative Computing, Video Conferencing, Video Archiving, Video Streaming**

I. Introduction

COMPUTING activities, including program sharing, document sharing and 24x7 data analysis operations, have become a central feature of modern engineering and research efforts. Engaging in informal interactions and sharing documents and data has proved to be an important part of effective collaboration. There is a need for collaborative tools that support connecting people so that they feel like they are working together on a daily, hourly, or even continuous basis. Messaging systems and presence awareness mechanisms can help make collaborations successful as they provide basic connectivity in a shared context that allows collaborators to hold spontaneous meetings.

IBM Lotus Sametime[1] is a middleware platform that enables unified communications and collaboration for enterprises. Lotus Sametime provides presence information, enterprise instant messaging, web conferencing, community collaboration, and telephony capabilities and integration. Sametime offers integration with Microsoft products such as Microsoft Office[2] and Microsoft Outlook[3]. Lotus Real-Time Collaboration Gateway allows exchange of messages between Sametime and other Instant Messaging applications such as Yahoo Messenger[4], AOL Instant Messenger[5] and Google Talk[6]. Sametime also provide a suite of tools and API's that enable the development of new applications as well as integration of Sametime features with other applications.

Lotus Sametime 8.5 is the latest member of the Lotus Sametime family. It introduces new features such as online meetings as well as enhanced telephony capabilities. Using the Sametime Meetings Participants can collaborate and interact remotely. With access available both from web based client and rich client, it can deliver more effective online meetings. One of the important features within Meetings is the Recording, where the participants can record the meetings for later playback. Lotus Sametime Meetings 8.5 leverages the capabilities of IBM Streaming Engine[8] for its recording functionality. Streaming Engine is a modular application which is capable of handling streaming functions.

This paper is organised as follows: Chapter 2 mentions about Streaming Engine and its high level architecture. In Chapter 3 a brief discussion on Lotus Sametime Meetings feature is presented. Meeting Recordings feature is mentioned in Chapter 4. In Chapter 5, a partner project, Agora[19] is discussed. This paper is concluded with Chapter 6, Conclusions and Future Directions.

II. IBM Streaming Engine

In recent years multimedia has become pervasive on personal computers. The most popular multimedia application, that almost all computer users are familiar with, is the so-called *player*. The best-known players include *Flash Player*[9], *Windows Media Player*[10], and *QuickTime Player*[11]. These players are typically capable of playing local multimedia files, but they are also able to connect over the network to a server in order to receive a streamed multimedia presentation. Apart from Video-On-Demand (VOD), i.e. serving stored files, these multimedia servers often can also capture a live source such as from a camera and stream this to connected players. Multimedia servers can be cascaded to realize scalability with respect to the number of users that can be connected to a live multimedia presentation.

These and other conceivable multimedia applications are typically dedicated programs that may share some of the underlying software. For example, a server that can be cascaded could use the same software for network input as a player. This modularity of the software has great advantages with respect to manageability of maintenance of the software.

IBM Streaming Engine[8] is a multimedia streaming server able to stream audio, video, images and text. It can serve both live and stored content. The streaming protocols are standards-based, such as the IETF[12] RTSP[13] protocol,

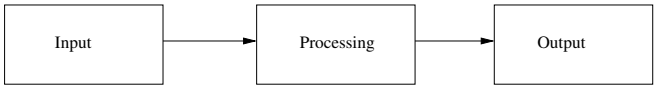

Fig. 1. General structure of a streaming application.

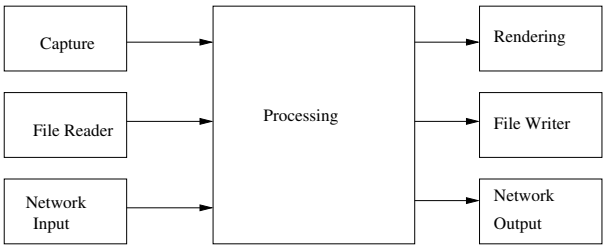

Fig. 2. A general streaming application with further specification of the inputs and outputs.

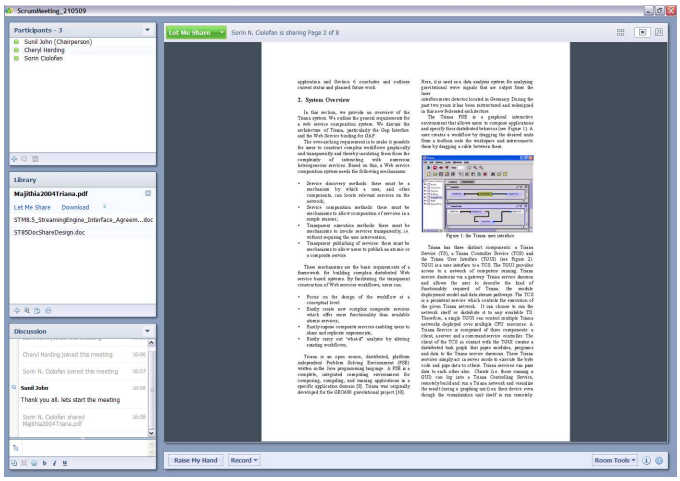

Fig. 3. Lotus Sametime Meeting workspace. A document is shared among the participants.

and the streamed content itself also adheres to standards such as MPEG-4 video and MP3 audio. In stored applications the server streams the content from a standard QuickTime/MPEG-4 file that does not require so-called *hint tracks*.

The basic architecture of Streaming Engine is simple and it is similar to a typical streaming application. A streaming application, be it live capture, VOD serving, playback, or other, can all have the same general structure, as shown in Figure 1. A player that is playing back from a local file has a local file reader as input, decoders as the processing units, and a renderer as output. Likewise, an IP camera has a captured signal for input, encoding for processing, while the output is a network output that for example uses the IETF protocols RTP/SDP/RTSP (see [15], [14] and [13]). Since this generalization can be fitted onto any existing multimedia streaming application, Streaming Engine adapts to this basic architecture.

Within the input module of Streaming Engine, three types of input are specified: Capture, Local File Reading and Network Input. Likewise, the output module also is subdivided into three types: Rendering, File Writing and Network Output. This is shown in Figure 2.

The connection between inputs and outputs is typically run via one or more processing elements. The most common are encoders and decoders where, for example, decoders are usually needed to connect a network input to a rendering output. Examples of other processing elements are: video down-sampling, speech-to-text conversion or face recognition.

By defining a certain data format between inputs, processing and outputs, it is possible to create new and/or custom inputs, processing elements and outputs. This makes Streaming Engine architecture extensible and configurable.

III. Lotus Sametime Meetings

Meetings is a new feature introduced with the Lotus Sametime 8.5 release. This feature has been introduced to improve the collaboration among the participants. A screenshot of Lotus Sametime 8.5 Meetings is shown in Figure 3. This feature essentially helps the users participate in meetings and collaborate. It is based on creation of virtual rooms, called meeting rooms, where the participants can collaboratively share documents, applications and discuss using the messaging system and audio. As depicted in Figure 3, the Meetings workspace has been divided into several sections for presenting and sharing meeting artifacts. Apart from uploading and sharing documents and applications, there are more advanced features available, like defining and sending polls, sharing URL's, saving or sending emails containing automatically generated transcripts from the *Discussions and Minutes* area. Annotations tools like highlighter (color configurable) and laser pointer are available to annotate and highlight the presenting documents. The Sametime Meetings is available for multiple OS platforms like Windows, Linux and Mac as it has been developed in Java.

IV. Meeting Recordings

The architectural specification of the recording functionality is presented in Figure 4. This feature leverages with Streaming Engine as well as with other Sametime Meeting components. Input entry into this component is Sametime Meetings 8.5 Rich Client. As specified in Chapter 2, Streaming Engine architecture consists of three components namely *Input*, *Processing Unit* and *Output*. These components are developed in C/C++ and provide an encapsulated JNI layer, through which a Java application can interact with Streaming Engine. Interaction between Sametime Meeting 8.5 components and Streaming Engine is shown if Figure 4. The Projector component of the Meetings module is responsible for rendering the shared application and document in the client window. The Notification Controller keeps track of the changes relate to the state of the data that captures. For example, if sound data is detected as a result of performing voice conversation, the

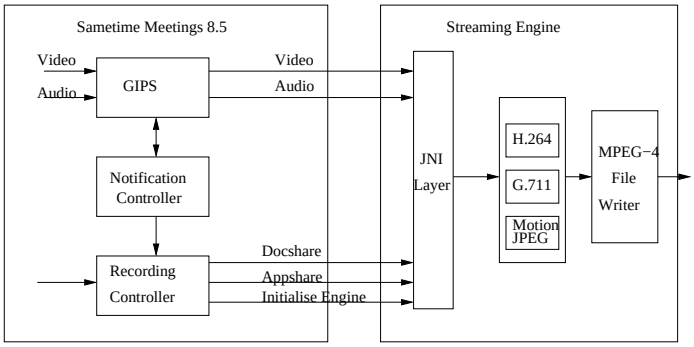

Fig. 4. Higher Level depiction of Meeting Recordings architecture.

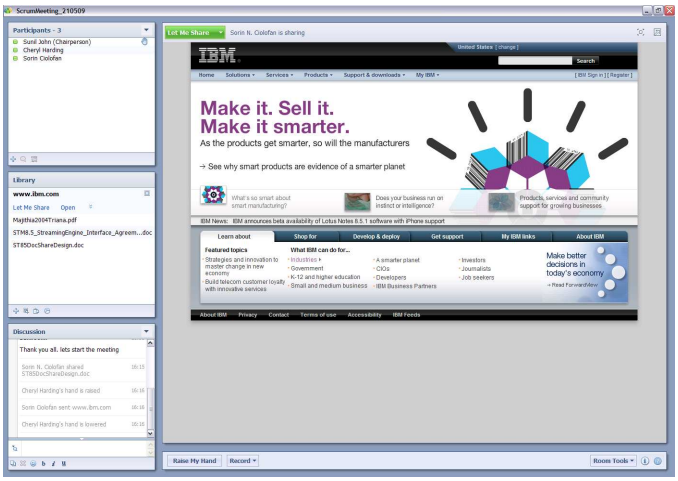

Fig. 5. Url sharing among the meeting participants.

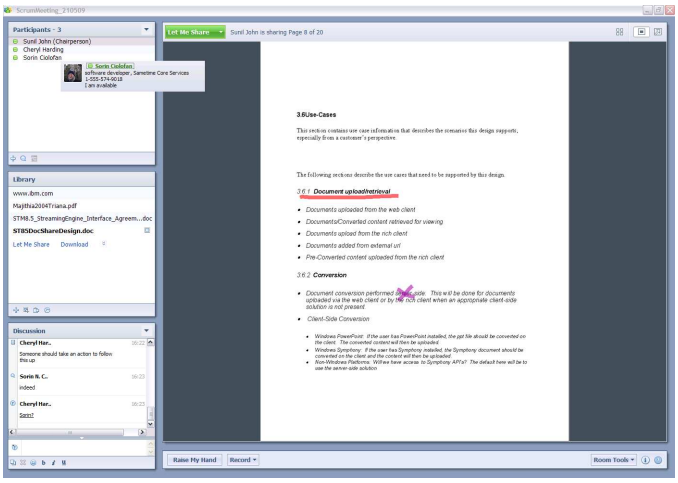

Fig. 6. Highlighter option to highlight the presentation.

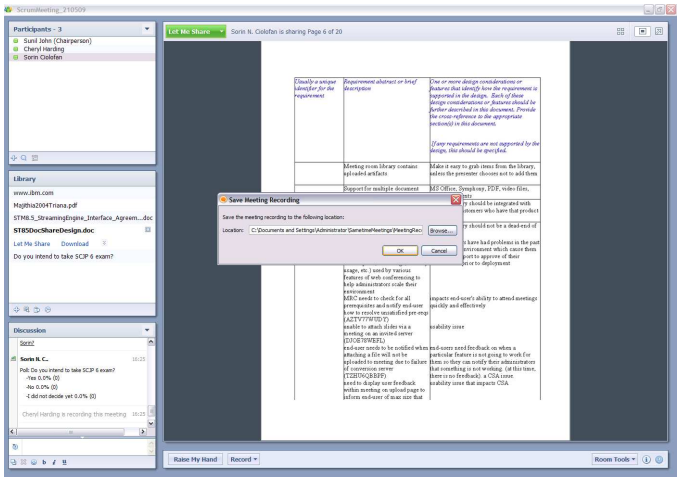

Fig. 7. Recording of a meeting. A message box to indicate the location of the saved movie file.

Notification Controller trigger events that needs to be handled further by the Recording Controller.

The GIPS Engine [7] is a platform-specific binary for the control and rendering of audio/video data on client workstations. GIPS is responsible for interfacing with hardware and providing an abstract API for managing media streams.

The recorded Meetings movie shows documents and applications shared by the participants during the meeting and includes the sound generated on clients' computers by microphones. The recording is in MPEG-4 format and consists of two tracks: one for the image and the other for the sound. The role of the Streaming Engine is to encode the streams of bytes corresponding to application/document share images as well as audio, and also to synchronize these tracks by using an unique time reference.

IBM Lotus Sametime [1] allows multiple remote parties to share documents and applications while conferencing using audio and video (Figures 5, 6 and 7). The Streaming Engine has been integrated into the Sametime product to record such *meetings*. The inputs to the Streaming Engine comprise of the audio and video streams plus screen captures of the shared applications or documents. The audio and video are recorded as is in their respective encoding formats (currently H.264 video and G.711 audio). The application/document window is input into the Streaming Engine in uncompressed format, where it is encoded using an H.264 video encoder. The application/document images may include mouse cursor and other dynamically visual items requiring a capture frequency in the order of a few frames per second. Therefore, using a video encoder like H.264 greatly reduces the file size over using image encoders like JPEG, while maintaining a good visual quality. The resulting recording is in the ISO Base File Format [16], which, depending on the content, is also known as QuickTime *mov* file, MPEG-4 file or Flash *f4v* file. The current Sametime collaboration meeting recordings can be played back by various standard players such as the QuickTime player[17] and the VLC media player[18].

Figures 8 and 9 show the comparison between video encoders H.264 and Motion JPEG when used for recording. The

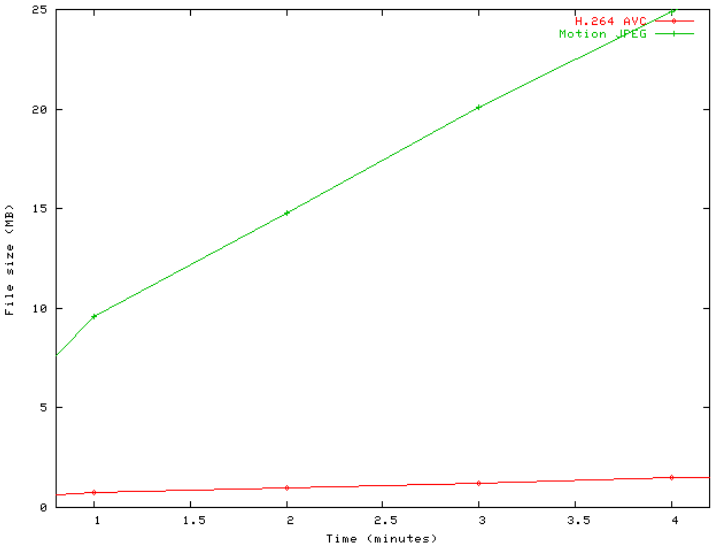

Fig. 8. Recording of a document share with annotations. Two video encoders—H.264 AVC and Motion JPEG—have been tested. As can be seen from the graph, H.264 produces smaller files.

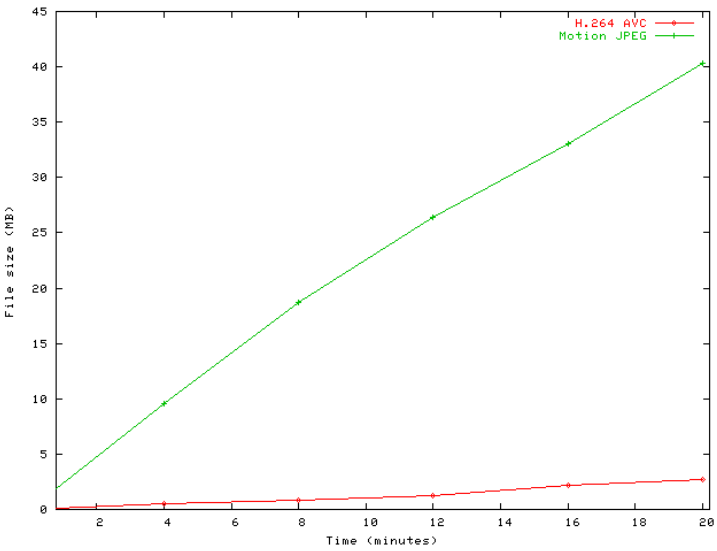

Fig. 9. Recording of Application and Document share. H.264 proved better performance.

experiments have been performed on IBM IntelliStation Z Pro workstation with 3.0 GHz Dual-Core Intel Xeon Processor running on Redhat Enterprise Linux 5.2. The first experiment is performed on document share with arbitrary annotations. The second experiment is a recording of simultaneous application and document share.

V. Agora Integration

Developed by the IBM Watson Research Center, Agora [19] is a collaborative media service whose prime goal is to draw upon the vast collective power of the crowd and skill sets found within the community, to *surface* ideas and information *buried* in meetings and expose this new dimension to viewers along the timeline. The web-based system enables users to upload meeting video or audio, automatically create transcriptions and attach metadata such as tags and comments along the timeline identifying and elaborating points and segments of interest. The video file becomes thus searchable, in the sense that a user can directly go to the clip of interest in the movie (based on annotations) rather than being forced to watch the entire movie in order to be able to extract the information he is interested in. The reuse of segments of the movie is also possible. The advantages that we envision from the use of Agora system within Meetings are: the possibility of persisting the recording in a centralized repository making it easy for later sharing with other users, adding *knowledge* to the movie (by the mean of various metadata), making the movie files searchable based on annotations and, not less important, allowing automatically transcription of the sound included in the video recording file in text format. These will make the recording of meetings more powerful in the context of collaboration.

VI. Conclusion

In this paper Collaborative Meetings experience using IBM Lotus Sametime as well as its recording capabilities is discussed. Sametime allows multiple remote parties to share documents and applications while conferencing using audio and video. IBM Streaming Engine is the backbone for providing the recording functionality to the Lotus Sametime. Streaming Engine supports multiple video encoders. Selecting an optimal encoder increases the recording performance. This paper also analyses the performance figures when choosing the video encoders. The recorded movie file is generally in MPEG-4 format, which can be playable using Apple Quicktime Player.

The future directions include the integration of the Lotus Sametime environment with Agora[19] which provides search and collaboration around recorded meeting artifacts. A Streaming Engine based custom media player should also be considered, which would allow avoiding dependency on other players. The other enhancement that can be developed is a Streaming Engine based File Server, where the recorded meetings files can be posted.

Acknowledgment

The support of IBM Dublin Software Laboratory and IBM Lotus Sametime Management teams are hereby acknowledged.

About the Authors

Sunil John: Sunil John is Technical Leader for the Lotus Sametime Meetings feature within IBM Dublin Software Laboratory. He received his PhD and MSc in Computer Science from University College Cork, Ireland and Bachelors degree in Computer Science from National Institute of Technology, Kerala, India. Dr. John joined IBM in 2006 and has been working in a number of areas including Service Oriented Architecture, High Performance Computing and Collaborative Computing.

Sorin Ciolofan: Sorin N. Ciolofan is a Software Developer in the IBM Dublin Software Laboratory, Sametime Core

Services team. He received in 2004 his MSc in Software Engineering from Politehnica University of Bucharest, Faculty of Automation Control and Computer Science. Prior joining IBM he has worked as a Software Engineer and R&D engineer in Java web applications development and Semantic Web.

Cheryl Anne Harding: Cheryl Anne Harding is a Software Developer in IBM Dublin Software Laboratory, Sametime Core Services team. She received her BSc in Computer Science from University College Dublin, Ireland. Prior joining Lotus Sametime team, she also worked as Extreme Blue Student with IBM Dublin Software Lab.

Peter Westerink: Peter Westerink is a research staff member at the IBM T. J. Watson Research center in Hawthorne, New York. He received his Ph.D. in 1989 in electrical engineering from the Delft University of Technology, the Netherlands, and joined IBM in 1995. Since then, Dr. Westerink has been working in the areas of video compression, multimedia streaming, standardization organizations (MPEG and ISMA), and Web services.

REFERENCES

[1] IBM Lotus Sametime, http://www.ibm.com/lotus/sametime
[2] Microsoft Office, http://office.microsoft.com
[3] Microsoft Outlook, http://www.microsoft.com/outlook
[4] Yahoo Messenger, http://messenger.yahoo.com
[5] AOL Instant Messenger, http://aol.com/aim
[6] Google Talk, http://www.google.com/talk
[7] Global IP Solutions, http://www.gipscorp.com
[8] IBM Streaming Engine, http://www.alphaworks.ibm.com/tech/streamingengine
[9] Adobe Flash Player, www.adobe.com
[10] Microsoft Windows Media, http://www.microsoft.com/windows/windowsmedia
[11] Apple QuickTime, http://www.apple.com/quicktime
[12] The Internet Engineering Task Force (IETF), www.ietf.org
[13] H. Schulzrinne, A. Rao, R. Lamphier, *Real Time Streaming Protocol (RTSP)*, RFC 2326, April 1998.
[14] M. Handley, V. Jacobson, *SDP: Session Description Protocol*, RFC 2327, April 1998.
[15] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, *RTP: A transport protocol for real-time applications*, RFC 1889, Jan 1996.
[16] ISO/IEC 14496-12:2008, *Information technology – Coding of audio-visual objects – Part 12: ISO base media file format*, http://www.iso.org
[17] Apple QuickTime, http://www.apple.com/quicktime
[18] VLC media player, the cross-platform media player and streaming server, http://www.videolan.org/vlc
[19] M. Topkara, B. Rogowitz, S. Wood and J. Boston, *Collaborative Editing of Micro-tags*, CHI Work-in-Progress (CHI-09), Boston, MA, April 2009