RC25038 (W1008-057) August 18, 2010
Computer Science

IBM Research Report

Tag Me While You Can: Making Online Recorded Meetings Shareable and Searchable

Mercan Topkara, Shimei Pan, Jennifer Lai, Stephen P. Wood, Jeff Boston
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Tag Me While You Can: Making Online Recorded Meetings Shareable and Searchable

ABSTRACT

In this paper we report on the design and implementation of a collaborative recorded meeting sharing system, with the goal of facilitating discovery and rapid access to information contained in recordings. The system, which we originally developed as a video sharing system behind the firewall - a sort of YouTube for the enterprise - evolved into a meeting capture, transcription, and annotation system. This evolution was driven by the high frequency and availability of recorded web-conference meetings with remote attendees joining via a common telephone bridge. The Collaborative Recorded Meetings system supports indexing and search not only in videos but also in any slides the attendees shared during the meeting. Recordings are automatically transcribed and indexed using speech-to-text technology and users can add annotations such as micro-tags (i.e. tags on segments of the meeting) and comments on the timeline. User generated tags can be shared via weblink with a colleague, who can then view just the segment of the recording associated with the tag. All attendees have the ability to edit the tags and transcription to improve the quality of the searchable data. We discuss the use of the system, along with findings from a 14 month video sharing trial within a large corporation.

1. INTRODUCTION

For better or for worse, meetings are an important and often principal part of a knowledge worker's day: decisions are made, information is shared and action items are defined. This information needs to be accessible after the meeting so that attendees can revisit what was discussed. Additionally, people who were not able to attend the meeting need to catch-up on what they missed.

It is undoubtedly as a result of the importance and preponderance of meetings that the field of research involving technologies which support meetings is so deep. The importance of this field is also recognized and funded by large government organizations, who have contributed to the meeting research agenda with projects such as the Augmented Multiparty Interaction (AMI) project [11], the Multimodal Meeting Manager project [13], and the Interactive Multimodal Information Management project [12], to name a few of the larger and better known ones. In parallel there have been focused efforts at Carnegie Mellon University and at Berkeley University to research methods of improving the automatic speech recognition for meetings [25, 3]. The goal of such research is to create a permanent, and often multimodal, record of the meeting [27] that can later be indexed and searched. The challenges are numerous and complex, involving multiple speakers, specialized vocabularies and spontaneous speech - with all of its associated problems. In many cases a dedicated room is used for the meeting. Meeting attendees are directed to the special room, which is outfitted with array microphones and cameras, and are often instructed to have 'simulated' meetings, in which they are given a scenario and a role to play. For some of the research projects [11] the speech is then manually transcribed by a human and annotated by researchers.

By contrast, our interest is in how computational tools can be used to seamlessly capture, process and share meetings that take place under everyday circumstances. While Lee et al. [29] introduced technology for a portable meeting recorder device that could be moved from room to room, it still required microphones to record the audio, and a special video machine to record the meeting. Today's meetings usually take place amongst members of a distributed team, using a bridge telephone conference and a web conference for screen sharing. The need for array microphones is satisfied by the audio from the telephone bridge, and the video can be automatically captured through the recording mechanism provided with most web-conferencing solutions [14, 15]. Web conferencing is the fastest growing tool amongst business users [16].

Our goal was to develop a solution that automatically processed such meetings, but was general enough so that it could also handle the case where a user wanted to upload a video recording of a meeting, for example a broadcast talk such as a lecture. By processed, we mean automatically transcribe and index the speech from the audio stream, index the slides, and capture the meeting metadata such as information on the host and attendees. In this case meeting processing can be viewed as only an intermediary (but necessary) step, since the focus of our research is around the collaborative aspects of sharing meeting artifacts, tagging meeting segments, and retrieving meaningful search results when searching across multiple data sources (transcription,

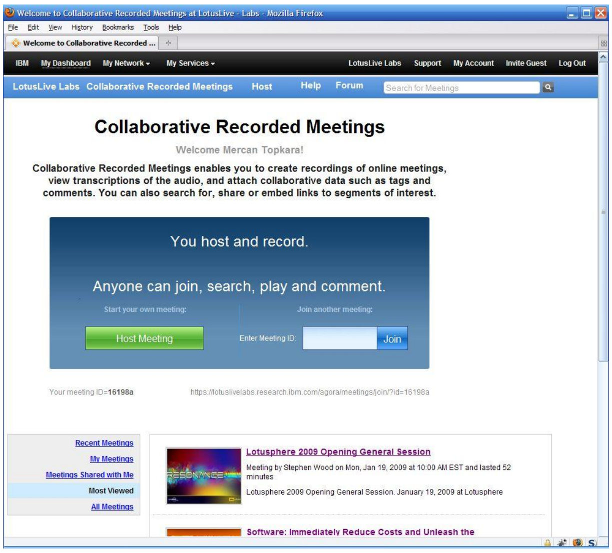

Figure 1: Welcome page for Collaborative Recorded Meetings (code name: Agora)

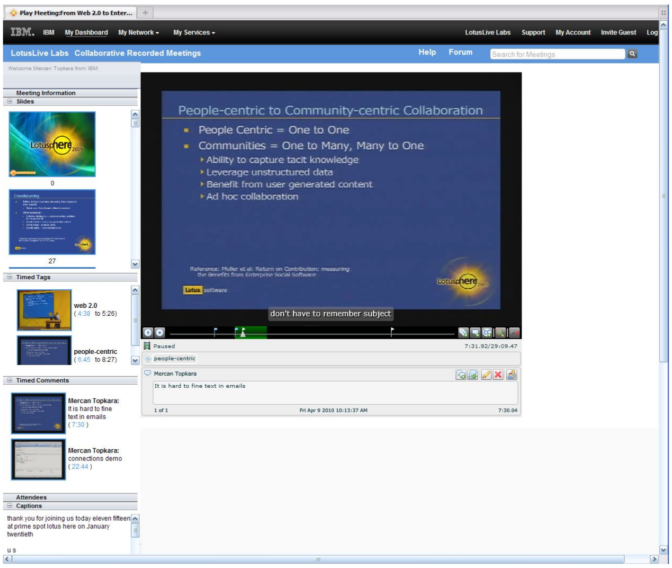

Figure 2: Agora Play Page

slides, tags, comments, metadata).

In this paper we present our research in this space of sharing and collaborating around automatically processed recorded meetings. In particular we describe our web-based solution, Collaborative Recorded Meetings system (code name: Agora), for searching and viewing recorded meetings, and introduce the notion of micro-tagging, i.e. tagging segments of meetings - with start and end times for the tag within the meeting recording - that can be shared via a weblink with collaborators. We also want to highlight that in order to develop a highly robust and scalable collaborative recorded meeting service for the enterprise, we adopt an open and distributed service-based architecture which can exploit and reuse existing and often sophisticated solutions for automatically analyzing content as well as beign open to several channels of user-data feed.

The paper is structured as follows: Section 2 provides a series of short motivational user scenarios that exemplify typical use cases for this system. In Section 3 we discuss how our research relates to prior work, focusing on systems whose goal is the viewing and sharing recorded meetings. We view meetings as a specific type of video recording, with specific use cases, and associated artifacts (e.g. slides). In Section 4 we describe the system, including its two versions; one for uploading and sharing videos of any type and the second, tailored specifically for recorded web conference meetings. In the following two sections we focus on the implementation aspects of core system components, and report on an extended user trial with the video sharing system. Lastly we present our plans for future work.

2. USE CASES

Playing and searching meetings to get familiar with a new team

Alice has just joined a new team. She would like to use the meeting service to get familiar with the vision of the team and a few ongoing projects. She came to the Agora site to do some research. As shown in Figure 1, Agora site provides users information on popularity of meeting recordings. She was given access to all the public meetings in the company and all the private team meetings. She started with playing a few popular meetings and skimming through the tags and comments to focus on heavily discussed parts. Figure 2 shows how Agora presents links to segments of a recording on the play page. Later she quickly went through the slide decks and listened to the discussions about a few interesting slides. As she knew the team and the jargons associated with the team projects better, she wanted to search the meeting repository to find video segments related to the project *Agora*. As shown in Figure 3 Agora search results page provides links to segments of meetings as well as visual queues to content when appropriate In this process, she found that Jeff was the person who knew Agora the most. To find more about Jeff and his work, Alice searches all the meetings that Jeff hosted or attended. After checking the search results, Alice understood that Jeff also worked on another project called Insight, which is closely related to Agora.

Annotating, editing and sharing metadata to promote a talk

Alice presented her Ph.D. work to her new team. Since she also wanted people outside of her team to know this work, she decided to make this meeting public in Agora. To help others find this talk, she tagged important parts of the talk with micro-tags. She also used comments on the timeline to give a few additional references. Since she found a few severe errors in the video transcript, she fixed them which makes it easier for others to understand the talk. This can also help users find relevant content using keyword-based page search on the html. As others visited her talk and put comments and tags, she can participate the discussion and respond with new comments. In addition, since she knew that Frank, a colleague from a different team would be interested in one of the topics covered in her talk, she tagged a few relevant segments and explicitly shared them with Frank using weblinks.

Collaboratively annotating important talks

Typically executive meetings are dense and may cover multiple topics. Since different teams may be interested in different topics, it is likely that each team may want to focus its discussion on different parts of the talk. These teams can collaboratively create and improve the metadata around different parts of the meeting. This collaboration, while not putting too much burden on each team individually, will create a full annotation of the meetings in the end.

Collaboratively editing tags to improve tag quality

Alice's manager George attended a meeting with a Vice President, and wanted to quickly inform his team on new deadlines for their deliverables. He added a few tags quickly so that his team would not need to watch the entire 60 minute recording. But he did not have enough time to perfect the tags. After receiving the weblinks of these tags in the email from George, the team started watching these segments. They found that the time stamps associated with the original tags were not precise and there were a few typos in the tags. In addition to the original project acronyms used by George, they also added the full name to facilitate keyword-based search. Some people also fixed the transcription to make sure the descriptions of the deliverables and the dates were accurate.

3. RELATED WORK

Our work is closely related to extensive efforts in both video tagging and sharing and collaborative meeting systems.

Video Tagging and Sharing There are many examples of manually annotating media. In the distance learning domain, there have been several systems where users can attach annotations to a time line of streaming media or index a video recording of a lecture with a synchronized presentation [7, 24, 26]. More recently, tagging and commenting on the video timeline has been implemented by several video services [5, 9, 10]. Some of these services also allow users to click on the frame of the video to attach a comment [6]. First example of explicitly tagging segments on streaming content was by Flavin [23] in the context of broadcast video. More recently, several news articles have referred to "micro-tagging" or "deep tagging" technologies, where users can select a segment on the video time-line [2, 8]. Similar to the above work, Agora allows users to add micro tags or timed comments to the video. Unlike the above systems where micro tags and comments were created manually, to facilitate meeting search and sharing, Agora also automatically add additional metadata on the video timeline such as meeting captions and presentation slides. Moreover, most meeting artifacts (e.g., tags and captions) in Agora are editable which has shown to be effective in improving annotation quality [31].

Collaborative Meeting Systems There are also extensive efforts on collaboratively creating and sharing meeting artifacts. Some of these efforts focused on enhancing the meeting room experiences. Among them, NoteLook [22], LiteMinutes [21] and MinuteAid [30] focused on collaborative multimedia note-taking in which different meeting artifacts such as video, audio and slide images were captured and incorporated as a part of the meeting minutes; eClass

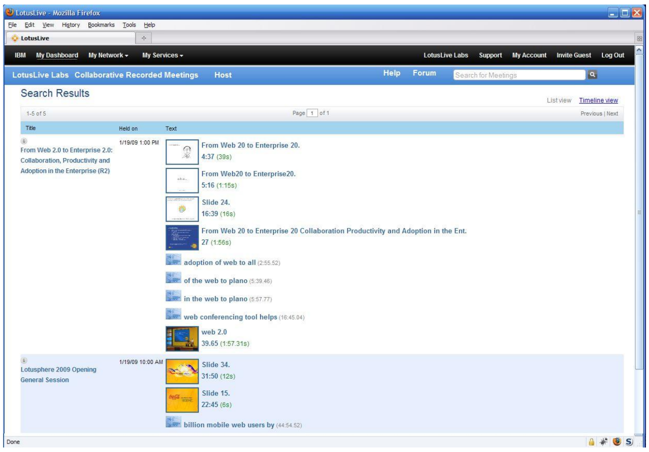

Figure 3: Agora Search Results Page

[17] is a system for visualizing multiple media streams, such as slide, audio and video streams, to enhance students' classroom experience; and WordPlay [28] is a collaborative multi-touch tabletop interface for generating and organizing ideas discussed during a meeting. Unlike these systems, Agora is a web-based collaborative meeting service that focuses on supporting geographically distributed users to attend remote meetings and to browse and search meeting artifacts.

Among the online conference systems that supports remote collaborations around meetings, Teamspace[24] is the closest to ours. It supports sharing and annotating slide presentations, creating bookmarks. In addition, all the session events (joining, leaving meeting) and user interactions are automatically recorded and time stamped by the server. They are subsequently used to index the meeting and are displayed on a time line to facilitate navigation. Unlike Teamspace that relies on an integrated solution for annotating, browsing and searching meeting artifacts , Agora adopts an open and distributed service-based solution to take advantage of existing enterprise solutions, such as the video search/sharing and the slide search/sharing services, to avoid duplicating existing efforts and to improve system scalability.

4. COLLABORATIVE VIDEO AND MEETING RECORDINGS SHARING

4.1 Design Principles

Enable collaboration on improving metadata

Our focus on enterprise environment where users do not have enough time to fully annotate meeting recordings but desire accurate annotations for effective information access makes collaborative metadata creation and revision appealing. Existing tag evaluation systems allowed users to rank a tag higher or add the same tag again to validate its appropriateness [1, 35], or let the automatic machine learning systems decide which tag is more relevant to a content [20, 32]. We employed a different approach which blends the design principles of wiki systems with tag evaluation systems. In our system, tags are editable as they are meant to describe the content and are useful in helping users find the content. Since our system also allows users to specify a time for a tagged segment, perfecting a tag becomes a more complicated task. Previous user studies showed that when users can edit tags, the quality of tags improves faster with fewer tags [31]. In addition, users participated in this study mentioned that they found the collaborative editing feature helpful as it gave them more control on fixing errors. The same design principle also allows users correct errors introduced by automated content analysis tools such as automated speech transcription. It was observed that some users only do selective editing to fix parts that are more important for them such as fixing the error in a project or a person's name.

Allowing users edit metadata on videos also helps automated content analysis systems leverage human input to improve their performance [34]. In our case, the annotations users performed on the videos can be used as training data for video segmentation and retrieval, or the corrections that users performed on the transcriptions can be used to improve the speech recognizer. Similarly the segments that have stirred most discussions can be used as training data for video and audio summarization.

Provide ability to search and share segments of media

It is important to allow users share segments from long video recordings, and also directly access the corresponding video segments from the search results. Rapidly accessing a relevant video segment is important, especially in an enterprise environment since it is common to record a long meeting that covers many topics in a meeting. Since many users may find most of the video content irrelevant, our system enable users to mark begin and end points for a tag indicating the topics related to the video segment. Our system also displays all the meeting segments that a slide was discussed since the presenter may come back to the same slide several times.

Open APIs for updating and retrieving data and content

Another design principle we followed was opening our system's APIs to external systems. This has made it easy to for others to integrate our system in their applications. For example, we have implemented *put* APIs to allow external components to add metadata (tags, comments, captions) to meeting videos. In addition, we also implemented search APIs to allow external systems search and retrieve data from our meeting repository. Through these APIs, external systems can query our database, fetch metadata from our system and add or update metadata on our system. In addition, since most external components Agora uses also follow the same design principle, these APIs enable them to maintain their independence while performing information integration through fluid two-way communication.

Being independent from how videos or recordings are created

The ubiquity of camcorders made it easy for anyone to record a video and share it with others. Similarly, the number of ways an online meeting can be hold and recorded varies widely. In order to cover the needs of a large group of users, we designed our system to allow users upload any standard format of video and audio recording. This has freed users from being restricted to a certain meeting recording system. The video can also be uploaded using an API call. This allows other external content sharing applications to integrate with our system without asking their users to visit our site to upload videos.

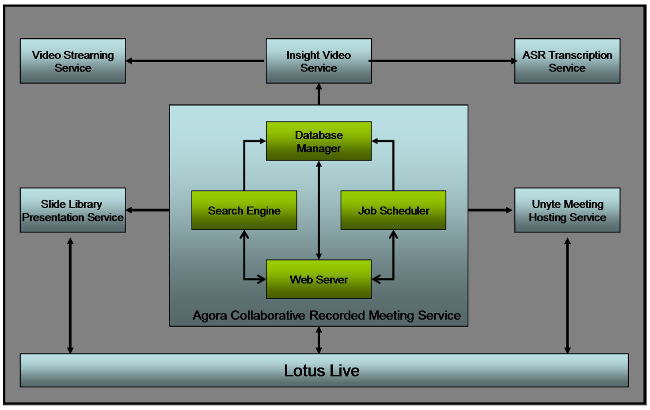

Figure 4: System Architecture.

Mashable user interface

Using our embeddable player, external social web sites can let their users playback the Agora videos with aligned metadata and add their own tags and comments on the timeline without leaving their web sites. The main reason that the external web sites can do this is because our main user interface components are mashable. For example, Agora's custom flash player can be controlled by the hosting HTML page. As a result, external web page designers can embed the Agora flash player in their own visualizations. They can also display the user inputs, display the metadata attached to the video timeline and control the Agora player all within the visualizations.

5. SYSTEM ARCHITECTURE

5.1 System Overview

Agora currently adopts an open and distributed service-based architecture. Agora itself is implemented as a *Lotus Live* service. It provides a set of APIs to help its clients create and search meetings on Agora. It also provides a web interface for users to directly create and search meetings from a web browser. Please note that *Lotus Live* is a commercial cloud-based offering that hosts a collection of integrated, online collaboration and social networking services for businesses.

Agora also relies on other services to implement its main functionalities. As shown in Fig. 4, Agora's key functionalities are based on three other services: the *Unyte meeting hosting service*, the *Insight video sharing service* and the *Slide Library presentation service*. The *Unyte meeting hosting service* is a commercial conference offering that allows remote users to join a meeting using a bridge telephone conference for audio and a web conference for screen sharing. The audio and screen sharing are automatically recorded by Unyte in a video file. It also automatically captures and broadcasts to its clients all the main meeting events such as slide switching and user joining events. It is worth mentioning that due to Agora's open API-based design, it is relatively easy for Agora to switch to a different meeting hosting service, as long as the new service can provide

APIs similar to those implemented by Unyte. The *Insight video sharing service* allows clients to upload a video, annotate the video with tags and comments, search the video based on audio transcripts, tags and comments and replay a video or video segment based on the search results. Similar to Agora, Insight also relies on other services to satisfy its needs. For example, it relies on the *ASR transcription service* to automatically generate captions from videos. It is also connected to the *Adobe Streaming Service* for video streaming and playing. Finally, the *Slide Library presentation service* allows clients to upload presentation slides in various formats (e.g., power point files and open document presentation files), add tags and comments for slides and search slides based on their titles and contents. Similar to Agora, both Unyte and the Slide Library are two independent Lotus live services.

In order to build the new Agora meeting service using existing services, we have developed four main modules for Agora: a *database manager*, a *job scheduler*, a *search engine* and a *web server*. The *database manager* stores the metadata associated with a meeting in a DB2 database. For example, it stores the information about when a meeting is created, who attends the meeting, what his/her roles are during the meeting (e.g., as a host, presenter or attendee), the IDs of all the slides presented in the meeting, the time each slide is presented, the associated video id and the status of the meeting (e.g., whether it has been marked for deletion). It also provides a set of APIs for other components to populate and search the database. The Agora *job scheduler* is the main component that interacts with other distributed services to construct and store meeting metadata and meeting artifacts in Agora. For example, it populates the Agora database with meeting metadata collected from the Unyte hosting service (e.g., when is a meeting created and who has attended the meeting) . It also uploads the meeting videos to the Insight video server and slides to the Slide Library when a meeting is over. Similar to the *job scheduler*, the Agora *search engine* also interacts with the other services, mainly through their search APIs, to produce the Agora search results. Finally, the Agora *web server* allows clients to create, browse and search meetings through a web browser. Currently, it is implemented as a Apache Tomcat server. Since both the job scheduler and the search engine are essential to the new Agora service, in Section 6.2 and 6.3, we discuss them in more details. We will also describe the Insight video sharing service more in Section 6.1 since the main ideas of Agora are evolved from Insight.

Before we describe the main Agora components in detail, we briefly explain how the Insight video sharing service is evolved into Agora, a collaborative recorded meeting service.

5.2 System Evolution

Even though enterprises have long been hosting large video repositories, little has been done to let employees use them practically. This work initially focused on developing a video sharing system that is tailored towards enterprises. The system we built, InSight, allows users to comment and tag video segments and share them with others via weblinks. It also allows users to edit the tags in order to improve their quality.

InSight is designed to be open so that automated content analysis tools, such as speech transcription tools [18], can be plugged into the system easily. To do this, Insight uploads videos to a content analysis service over its upload

Figure 5: InSight Player

APIs and receives the analysis results. Then the analysis results are displayed in the Insight user interface. Moreover, most Insight functionality (such as adding tags and comments, uploading videos, and search) is available through external API calls. This makes it easy for other application developers to integrate InSight or its components such as the video player, in their application. This open API-based design allows us easily extend around Insight to build a full recorded meeting service.

We released an internal version of InSight with captions on February 2009. We will report more about our experience with this release in Section 7.

Recently, we shifted our focus towards recorded meetings since meeting is one of the most shared type of videos within an enterprise. There are many ways we can leverage the openness of our video sharing system to enhance the collaborations around meeting recordings. In this process, InSight became one of the subcomponents for meeting video storage and sharing. In addition, we also added Slide Library [19] for storing and sharing slides presented at the meeting and Unyte for hosting and joining online meetings. We built our new meeting sharing service on Lotuslive [4] to take advantage of some common functionalities provided by LotusLive, such as access control, which are needed by different LotusLive applications.

6. SYSTEM COMPONENTS

6.1 Video Sharing

The version of the InSight system that was used as a sub component of Agora is shown in Figure 6. It includes a custom Adobe Flash/Flex based client player running in a web

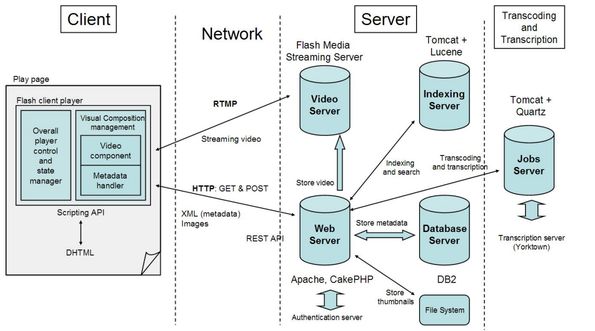

Figure 6: InSight Architecture

browser and a server application for storing and retrieving video and metadata (e.g., micro tags). The server application was built using an HTTPs server (Apache), a database (DB2) and server-side scripting (PHP). Communication between the client and server is done using XML exchanged over HTTPs connections allowing tags to be created, edited and deleted. Each modification to a tag is recorded and the change history is tracked. Video recordings for meetings are stored in the filing system and they can be played back over RTMP from an Adobe streaming media server. Using a streaming media server (Adobe) allows better precision and seeking to arbitrary locations. The videos are converted to flv format.

In addition to tags, InSight supports captions in multiple languages. The captions are very similar to tags, except in a caption, several words are grouped and assigned a begin and end time, according to the times returned in the transcription. Like micro-tags, captions can be searched and edited by viewers . Captions are automatically generated using a transcription server via REST APIs [18]. The editing feature provides the users the ability to fix errors in the transcription results. The transcription errors usually occur on out-of-vocabulary words such as person or product names.

The server generates RSS feeds to return the results such as video and meta data to its clients. The client can be embedded on any web page to allow users modify the metadata locally.

We used a Quartz job scheduler in order to manage several jobs in parallel. The Job scheduler starts the transcoding and transcription process. It also uses Lucene to create a search index for tags, comments and transcription.

6.2 Automatic Metadata Generation

Agora system automatically creates some of the metadata on meeting recordings before it is available for viewing and sharing. *Job scheduler* manages all asynchronous processes during a meeting recording creation at Agora repository. It is one of the main components that make it possible for Agora system to take advantage of various existing capture, analysis and content sharing services. In addition to the tags and comments added by users, Agora stores transcription, slide text and meeting attendee information as metadata attached to recordings. In this system, all metadata is timestamped only with the exception of the meeting summary and title information provided by the owner of the recording.

Agora job scheduler performs the following high level actions: (i) listening to the meeting to record during meeting events; (ii) retrieving the encoded video and shared slides when the meeting is stopped; (iii) pushing the shared content to several subcomponents and updating the Agora database server as output comes back from these components' analysis. We will explain these steps in more detail in the following paragraphs.

Agora job scheduler is notified when a user initiates a "host meeting" event at Agora web site. At this time, job scheduler collects the required credentials to authenticate itself to monitor the ongoing meeting. Through the APIs provided by Sametime Unyte meeting service, job scheduler is able to monitor all during meeting events such as participant actions, presenter changes, public chat and slide change events. We ask the user to upload the slide deck through Agora interface in order to get a copy of the ppt file. All other user interaction during a meeting is handled directly through the Unyte frame embedded in our host/join pages.

When the host stops the Unyte meeting, job scheduler component retrieves the .mov file created for the recording as well as the ppt file posted on the Agora server. Job scheduler uploads video file to Insight and uploads shared slides (slides that were not displayed during the meeting are discarded from the ppt deck) to Slide Library repository. Both of these systems have several format conversion and data extraction analysis in place which takes various times depending on the size and content of the videos and slides. Job scheduler follows the progress of these analysis through the event notifications received from these services. These subcomponents can perform their analysis in several steps, for example InSight performs transcoding and transcription of the video independently, completion of each event are notified to the job scheduler.

Job scheduler also handles the correct population of pointers to the hosted content in the Agora database. For example, job scheduler creates an entry for each slide that was shared during the meeting with the right slide id pointing to Slide Library and the time(s) this slide was on the screen. Similarly, it populates user data with the times they performed a certain action such as "join meeting", "present".

6.3 The Agora Search Engine

The Agora search engine is another component that integrates the functionalities of various independent services to create a new service. It allows users to identify relevant meeting segments based on the associated meeting artifacts such as audio transcripts, slides, tags and comments. Since all the information is time-stamped, users can directly click on the search results to replay the video segments. The input to the search engine is a user's search request. Currently, Agora supports keyword-based search over meeting transcripts, titles, summaries, tags, comments and slides. It also supports meeting metadata-based search such as "find all the meetings within the last 7 days" or "find all the meetings in which Jeff is a presenter". The output is a list of ranked meeting objects in XML. The search results can be rendered in different client environments such as a web browser or a Lotus Notes plug-in.

There are three modules in the Agora search engine: a *query translator*, a *search result composer* and a *search result evaluator*.

The *query translator* maps an Agora search request into several data source-specific queries. Here, "data source" means different data repositories that store different meeting artifacts such as videos and slides. For example, given a user query "Find all the meetings in which Jeff or Steve presented Agora and Insight". The corresponding Agora query is a conjunction of two main constraints. First, a keyword constraint that requires both the word "Agora" AND the word "Insight" appear in the meeting. The second constraint limits the search result to those either Jeff OR Steve was a presenter". When the Agora search engine receives the query, it first translates the query into multiple data source-specific queries such as a video search query, a slide search query and an Agora metadata query. The main challenge here is to ensure that the resulting queries can maintain a high degree of query fidelity without scarifying the search performance. Here, "query fidelity" means how faithful or accurate the resulting queries can convey the semantics of the original query and "search performance" measures how fast Agora returns search results. Since both data retrieval and search result merging are time consuming, to ensure the best search performance, the optimal solution would require the query translator to minimize both the number of search queries issued to different data sources and the number of search results returned from each query. In the above example, to minimize the number of queries issued to different data sources, Agora would need to issue a keyword query that finds all the videos with the word "Agora" OR the word "Insight". It would need to issue another query to get all the slides containing "Agora" OR "Insight". Then, after both results returned, the system would parse each result and select meetings that contain both the word "Agora" AND the word "Insight". Since the search results are not indexed, parsing and filtering can be inefficient. Moreover, since the keyword constraints may include arbitrary number of keywords combined by different boolean operators such as AND, OR and NOT, it would be tricky to optimize the set of data source-specific queries that maximize search performance while maintain query fidelity. Our current implementation favors search performance over maintaining the exact query semantics. For example, it only issues at most one query for each data source. To handle the keyword constraint above, for example, Agora would issue an Insight query to find videos containing both "Agora" AND "Insight". It issues a Slide Library query to find slides containing both "Agora" AND "Insight". Even though this approach may fail to retrieve some meetings such as those that have "Agora" in the video and "Insight" in the slides, it is efficient since it minimizes the number of queries issued to different data sources and there is no needs to parse the search results.

Based on the search results from various data sources, Agora *search result composer* dynamically creates a list of meeting objects that meet the search criteria. A data source-specific search result would need to meet four criteria in order to be included in the Agora search results, (1) It belongs to a meeting (not every video in Insight or every slide in the Slide Library belongs to a meeting) (2) It belongs to a meeting that is valid (e.g., not those tagged as "to be deleted" or "not ready") (3) It belongs to a meeting that the user has access to. (A user can access a meeting if he attends the meeting, or the meeting is explicitly shared with him by another person who has access to the meeting, or it is a public meeting). (4) When there are additional metadata constraints in the query, the result should belong to a meeting that stratifies all the metadata constraints. At the end of this step, a list of meeting objects are created, each contains all the data source-specific search results such as retrieved captions, slides and tags, related to the current meeting.

Given the list of meeting objects constructed by the *search result composer*, the *search result evaluator* first determines the relevance of a meeting object to the search query and then ranks them according to its relevance to the query. Comparing, composing and ranking new search results based on the search results from distributed data sources are notoriously difficult [33]. First, different data sources may return different kinds of information in their search results. For example, some may just return a ranked list. Others may also include a search score. Second, the search ranks and the search scores are not directly comparable across different data sources (e.g., we can not say whether the highest ranking slide in the slide search results is more or less relevant than a lower ranking transcript in the video search results.) One solution to these problems is to use a centralized meeting index for Agora. Unlike federated search in which each data source-specific service may be uncooperative, Agora has access to all the meeting content when a meeting and its associated artifacts are created. For example, when a user uploads slides to Agora to be presented in a meeting, it is possible for Agora to extract the textual content in each slide. Similarly, when the transcript of a video is created by the transcription server, Agora can also index the words in the transcript. Thus, it is possible to build a centralized search index for all the artifacts associated with a meeting. During search, the rank/score of a meeting is simply determined based on the centralized meeting repository. Thus, there is no needs for comparing and composing search ranks/scores from different data sources. This approach however, suffers some main drawbacks. For example, to build a centralized meeting repository, Agora would need to create and store its own meeting index. This duplicates the efforts by the other services in creating data source-specific search indexes. Moreover, depending on the APIs between Agora and the other services, Agora may not be able to receive every data source-specific internal event in real time. As a result, the centralized search index may be out of sync with the content of a data source. For example, after a user adds a comment to a video, the video server may not report this event immediately to Agora. Since the centralized meeting repository has not updated its index with this new information, the newly added comment will not show up in the Agora search results.

To take advantages of centralized indexing while avoid its deficiency, currently we adopt a semi-centralized search process. It has two phases: an off-line training phase and an online evaluation phase. During the offline training phase, Agora builds a centralized meeting repository. During training, for each search query, Agora automatically creates training instances consisting of retrieved data and the associated ranks/scores from each data source and the overall Agora search rank/score from the centralized index. Based on these training instances, Agora needs to determine the best strategy to combine the ranks/scores from each data source in order to predict the overall rank/score of a meeting. Currently, Agora learns a linear regression model based on the training instances. This model is then used in the online phase to determine the search score for a meeting. This model is updated periodically to keep in sync with the content of all the data sources.

7. USER TRIAL

We observed the InSight system for 14 months (February 2009 and March 2009) after it was released and announced within a large corporation. We have observed on the average 434 unique IP visits per month. 265 videos were shared in total. We saw the highest turn-out on July 2009 right after our site was used to share the recordings of the talks and panels at an important company event. Figure 7 shows the visit distribution over the months. We have observed an increase on the monthly visit after January 2010 where we attended a large event within the company and promoted our system.

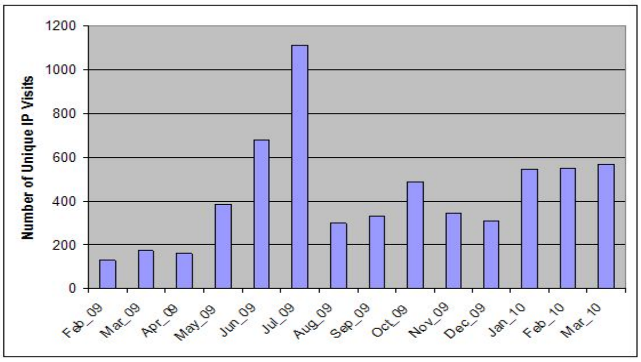

Figure 7: Number of Unique Visitors at InSight

A significant amount of content request to InSight was from other domains. 11% of the traffic at InSight originated through a referral from a different domain. We have observed that a significant amount of these content requests come from pages that embedded a video to their HTML page using our custom player. The referral that caused the top number of hits in a month (25546) among the hits coming from external referrals is through a news page at a company portal where editors embedded a podcast for their yearly review. Figure 8 shows that the bandwidth of this traffic changes significantly from month to month. This graph also shows that following the design principle of creating a mashable user interface provides flexibility for users on how to share their content.

We observed that on the average only 3.5% of monthly activity at our site was due to search queries. Other activities on our site include playing a video, browsing, adding metadata, etc. This is lower than our expectations, however as also seen on Figure 9, there is a trend towards higher ratio of search activity. We expect search functionality to be used more often as the total number of videos at InSight repository increases. We also expect that there will be higher search activity at Agora recorded meeting repository, since the recordings is expected to be longer as a typical web conference meeting takes at least 30 minutes. The median duration of top 30 most viewed videos at InSight is 9 minutes, and the average duration of these videos is 30 minutes.

We have also analyzed the distribution of play page requests to see if weblinks for playing segments were used by our users. Figure 10 shows that the percentage of segment play requests dropped significantly after the first few months of releasing our system. We suspect that this might be due

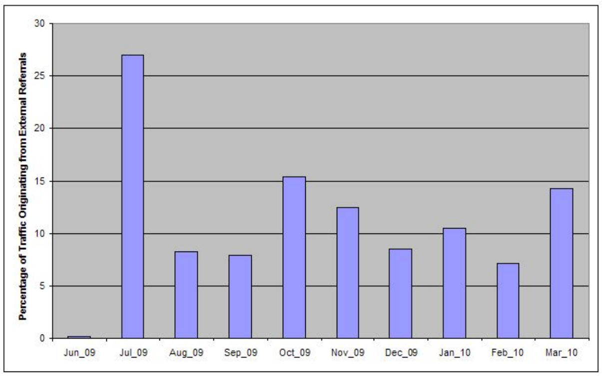

Figure 8: Percentage of external page referrals to InSight between June 2009 and March 2010 (There were no external referrals between February and May of 2009).

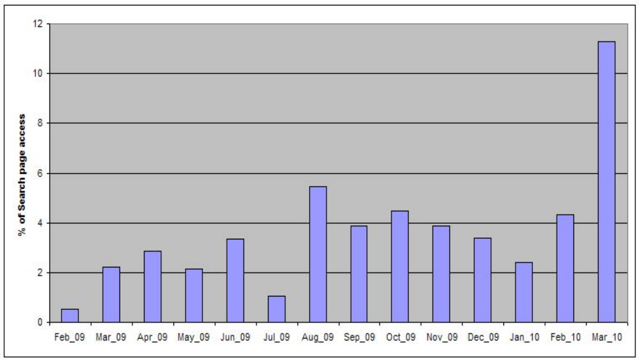

Figure 9: Number of Search Queries at InSight

to the increase in the variety of videos we host and the visitors. As Figure 8 also shows a high percentage of our traffic is via weblinks and embedded players, which might be referring users to the full play link or using the embedded player to embed the full video.

Figure 11 shows the metadata creation activity performed by users. These numbers do not include the automatic creation of captions. We observed that the edit activity was very high on September 2009 on a particular video that was promoted highly within a department by their communications team. This supports our motivation for expecting users to higher incentive for improving the metadata on recorded meeting videos when there is a business need. We expect to have higher interest from users to improve the metadata around recorded meetings that they would like to make discoverable for others and for themselves.

We also observed that the content creation activity at InSight was performed by a small number of users, that share most of the videos. 269 users logged into our system throughout the course of 14 months. We have observed that the activity of users follow a power-law distribution, where a small number of users contributed a large portion of the content. For example, 93% of videos were uploaded by 15% of users. Similarly, 97% of tags were added by 8% of users.

InSight was designed to be open to all visitors within the company to become a portal of interaction around video sharing within the company firewall. Any user can play and

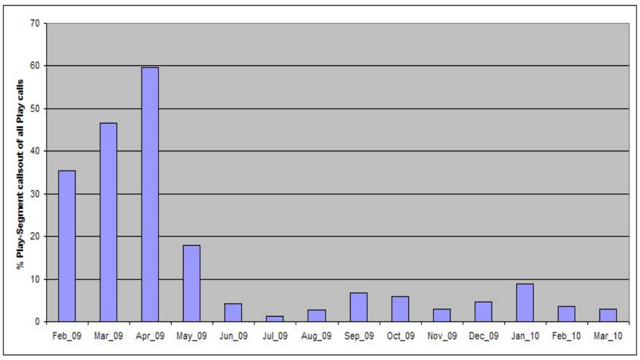

Figure 10: The percentage of play page calls that directly played a segment versus total number of play page calls that also included the requests for playing the full video from beginning

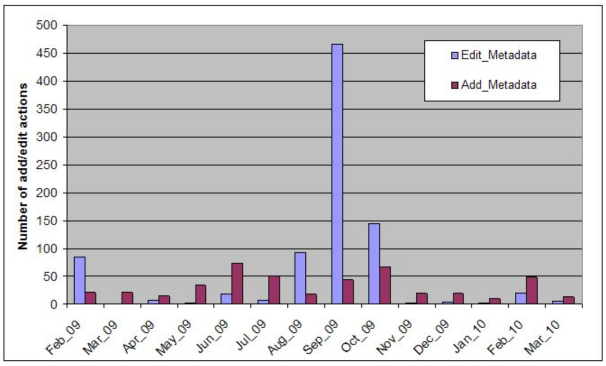

Figure 11: Number of add/edit actions on tags, comments and captions by users at InSight

search videos without logging in. However, in order to be able to track user activity around metadata (tags, comments and captions) creation, we put in place the requirement to login to add or edit metadata in addition to the requirement of logging in for uploading a new video. Our login mechanism is based on OpenId, which requires our users to go through a 3 step process to authenticate. We suspect that this might be another reason that even though we see very high number of visits from unique IPs, we do not see that numbers to be reflected in metadata creation. Agora is hosted in LotusLive environment and the authentication is based on single sign on, where users can navigate between all services without to need for entering credentials again.

Figure 12 shows a significant amount of content requests were to play tutorial videos and to view help pages. We believe that if we improve our user interface to make it easier for users to contribute content to our web page, we will start seeing higher traffic around adding and editing the metadata on videos. We discuss more about our future work plans in Section 8.

8. DISCUSSION AND FUTURE WORK

Comparing to InSight, we have added many new features to Agora in order to increase user engagement and contribution. We expect to see more search and share activities in Agora for several reasons: (i) meetings are longer than

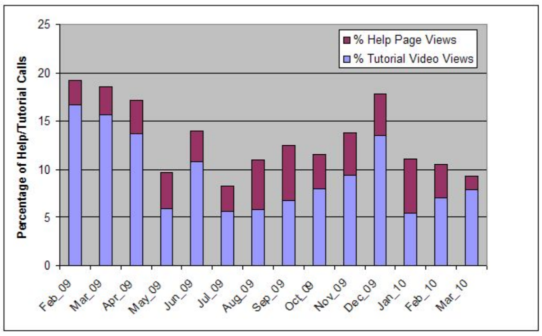

Figure 12: Percentage of help page visits and tutorial video views compared to total number of page visits at InSight between February 2009 and March 2010

typical videos shared online; (ii) users have business needs to access information buried in monolith blocks of meetings; (iii) Agora can search over richer data sources(e.g., slides, attendee information). Other new features will also improve search and sharing in Agora such as (i) sharing a meeting with attendees in Agora is automatic; (ii) Agora is a part of the Lotus Live platform that hosts other social networking services such as e-mail and blog.

Agora will be available to users at the end of April 2010. We plan to track user activities to verify the above hypothesis. In the meanwhile, we can further improve user engagement by recommending relevant meetings on the play page. We also plan to group meetings from the same theme together to improve meeting organization.

Based on what we learned from InSight, we realized that we need to make the interface for metadata creating and sharing easier to use. We plan to allow users to add tags (and comments) to an ongoing meeting. These tags will be saved and later made available as metadata on the recorded meeting. We are also working on improving the user interface of our video player and our play page in order to provide better guidance for users to add metadata and navigate within a shared meeting.

Another feature that we are working on is to improve search results. As we have explained in Section 6.3, performing federated search on different data resources is a hard problem. In our case, we see several things we can implement to improve the quality of search results. For example, transcriptions are valuable since they are automatically created and they are almost always available. Slides however may not be available in a meeting. Due to the limitations of speech recognition technologies, the quality of transcriptions varies significantly from meeting to meeting. Even within a meeting, it varies from speaker to speaker. To better evaluate the search results from video transcripts, we plan to use the confidence scores returned by a speech recognizer to better predict the search scores. We are also working on improving the transcription quality using the slides uploaded during a meeting.

9. CONCLUSION

In this paper, we have shared the design and implementation of a collaborative recorded meetings system, Agora, and also that of its predecessor, InSight. We started this work by first building InSight and deploying it in a large corporation. Later, we shifted our focus and build a collaboration system that is tailored more towards recorded online meetings, where remote users participate a meeting through a web interface.

While building these systems we followed five main design principles: (i) enabling user collaboration on improving metadata; (ii) providing users the ability to search and share segments of media; (iii) providing open APIs for updating and retrieving data stored in our system; (iv) being independent of how videos or meeting recordings are created; (v) providing a mashable user interface to enable other social networking systems to embed our system.

We also summarized some user statistics from Insight. We want to evaluate whether the above design principles have improved user experience (Section 7). We found that users reacted very positively on being able to embed our player in their own web pages. They also appreciate that they can mash their user interface with InSight features. In one particular case, Insight is integrated with a sales team portal to enable their users upload, share and view videos directly from their interfaces.

We also learned a few lessons from these experiences. We realized that we need to improve the current user interface for adding and editing tags, comments and captions. Even though search and sharing on Insight are quite limited, we expect this to change in Agora as meetng recordings will be longer and users will have higher business needs for effective access to meeting data.

10. REFERENCES

[1] http://www.amazon.com.

[2] http://www.clickz.com/showPage.html?page=3624735.

[3] http://www.icsi.berkeley.edu/speech/mr/mtgrcdr.html.

[4] http://www.lotuslive.com.

[5] http://www.motionbox.com.

[6] http://www.nicovideo.jp.

[7] http://www.sonicfoundry.com/mediasite/.

[8] http://www.techcrunch.com/2006/10/01/all-the-cool-kids-are-deep-tagging/.

[9] http://www.veotag.com.

[10] http://www.viddler.com.

[11] www.amiproject.org.

[12] www.im2.ch.

[13] www.m4project.org.

[14] www.webdialogs.com.

[15] www.webex.com.

[16] Meetings in America V: Meeting of the Minds. In *MCI Whitepaper*, 2003.

[17] G. D. Abowd, C. G. Atkeson, J. A. Brotherton, T. Enqvist, P. Gulley, and J. LeMon. Investigating the capture, integration and access problem of ubiquitous computing in an educational setting. In *CHI*, 1998.

[18] K. Bain, S. Basson, A. Faisman, and D. Kanevsky. Accessibility, transcription, and access everywhere. *IBM Systems Journal*, 2005.

[19] C. Y. Chi, R. Konuru, W. P. Xiao, and D. Yeh. Bringing collaborative edting of open document format (odf) documents to the web. In *XML*, 2007.

[20] E. H. Chi and T. Mytkowicz. Understanding the efficiency of social tagging systems using information theory. In *Hypertext*, 2008.

[21] P. Chiu, J. Boreczky, A. Girgensohn, and D. Kimber. Liteminutes: an internet-based system for multimedia meeting minutes. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, 2001.

[22] P. Chiu, A. Kapuskar, S. Reitmeier, and L. Wilcox. Meeting capture in a media enriched conference room. In *Multimedia Magazine*, 1999.

[23] R. A. Flavin. Control of a system for processing a stream of information based on information content, July 24, 2007.

[24] W. Geyer, H. Richter, L. Fuchs, T. Frauenhofer, S. Daijavad, and S. Poltrock. A team collaboration space supporting capture and access of virtual meetings. In *ACM SIGGROUP Conference on Supporting Group Work*, pages 188–196, 2001.

[25] R. Gross, M. Bett, H. Yue, X. J. Zhu, Y. Pan, J. Yang, and A. Waibel. Towards a multimodal meeting record. In *ICME*, 2000.

[26] S. Houde, R. Bellamy, and L. Leahy. In search of design principles for tools and practices to support communication within a learning community. *SIGCHI Bulletin*, 30(2):113–118, 1998.

[27] D. Huggins-Daines and A. Rudnicky. Implicitly supervised language model adaptation for meeting transcription. In *NAACL HLT*, pages 73–76, 2007.

[28] S. Hunter and P. Maes. Wordplay: A table-top interface for collaborative brainstorming and decision making. In *3rd IEEE Int. Workshop on Horizontal Interactive Human Computer Systems*, 2008.

[29] D. Lee, B. Erol, J. Graham, J. Hull, and N. Murata. Portable meeting recorded. In *ACM Multimedia*, pages 493–502, 2002.

[30] D. Lee, J. J. Hull, B. Erol, and J. Graham. Minuteaid: multimedia note-taking in an intelligent meeting room. In *IEEE ICME*, 2004.

[31] M. Topkara and B. Rogowitz and S. Wood and J. Boston. Collaborative editing of micro-tags. In *CHI extended abstracts*, 2009.

[32] S. Sen, J. Vig, and J. Riedl. Learning to recognize valuable tags. In *Intelligent User Interfaces*, 2009.

[33] L. Si and J. Callan. A semisupervised learning method to merge search engine results. *ACM Trans. Inf. Syst.*

[34] L. von Ahn and L. Dabbish. Designing games with a purpose. *Communications ACM*, 51(8), 2008.

[35] Z. Xu, Y. Fu, J. Mao, and D. Su. Towards the semantic web: Collaborative tag suggestions. In *Collaborative Web Tagging Workshop at International World Wide Web Conference*, 2006.