Building Digital Project Rooms for Web Meetings

Laurent Denoue, Scott Carter, Andreas Girgensohn, Matthew Cooper
FX Palo Alto Laboratory
3174 Porter Dr.
Palo Alto, CA, 94304 USA
{denoue, carter, andreasg, cooper} @fxpal.com

ABSTRACT

Distributed teams must co-ordinate a variety of tasks. To do so they need to be able to create, share, and annotate documents as well as discuss plans and goals. Many workflow tools support document sharing, while other tools support videoconferencing, however there exists little support for connecting the two. In this work we describe a system that allows users to share and markup content during web meetings. This shared content can provide important conversational props within the context of a meeting; it can also help users review archived meetings. Users can also extract content from meetings directly into their personal notes or other workflow tools.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Conferencing

Keywords

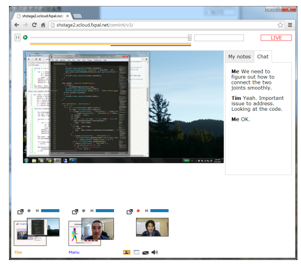
Web meeting, video, annotation

1. INTRODUCTION

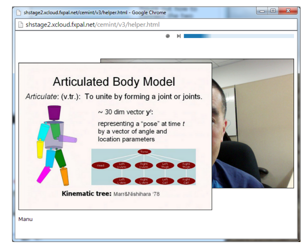
Project rooms are dedicated spaces that allow participants to revisit content in a series of meetings over weeks or months [1]. They involve the "creation, editing, and persistence of flexible documents" to support long-term collaboration around complex tasks. Physical project rooms typically include large shared spaces that a collocated design team can visually scan and utilize with ease. While aspects of the physical design space are invaluable, they increasingly do not align with the reality of modern knowledge work in which participants are distributed. Furthermore many smaller organizations do not have enough real estate to allow them to devote an entire room to a single project for weeks on end.

DocEng '14, September 16 - 19 2014, Fort Collins, CO, USA

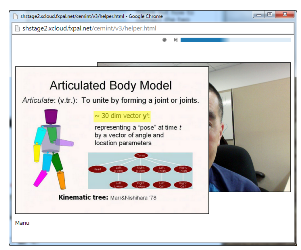
ACM 978-1-4503-2949-1/14/09 ...$15.00.
http://dx.doi.org/10.1145/2644866.2644889.



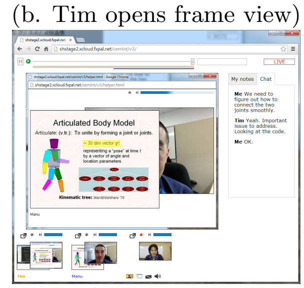
(a. Sue's main view)



(b. Tim opens frame view)



(c. Tim's frame view)



(d. Sue's main view)

Figure 1: An interactive web meeting. (a) In the main view, the current user (Sue) is looking at Tim's currently active stream in the main view. She can also see Tim's and Manu's frame archive along the bottom. (b) Tim clicks on Manu's icon to open his frame archive in a new window. (c) Tim annotates a frame from Manu's archive and shares his annotation to the rest of the meeting participants. (d) Sue sees Tim's view updated in the main view.

In this work, we are interested in supporting purely *digital* project rooms – online spaces that allow a group of participants to create, view, annotate and edit documents in a flexible and persistent way. As described in [1], key components of such a room include shared, editable spaces, coordination documents (such as flow charts, todo lists, and other workflow tools), as well as co-working space. Online tools already exist that focus on editable digital spaces and coordination documents. For example, Trello[1] allows users to create a variety of workflow documents and to populate them with multimedia information. Other videoconferencing tools allow users to share, and discuss documents in a meeting environment (e.g., WebEx[2]). However, project

[1]http://trello.com/
[2]http://www.webex.com/

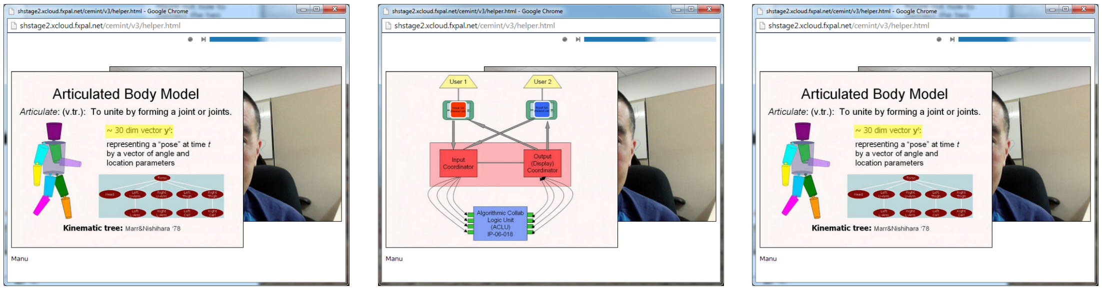


Figure 2: Annotations in our system are linked to underlying content. Here, Tim cycles through Manu's frame archive in a pop-up window (Manu is currently sharing his webcam, which appears in the background in the upper-right). Tim first views a slide he previously annotated (left) and then swipes to view the next slide in the deck (center). Next he swipes back to the first slide (right).

rooms should support both *synchronous* and *asynchronous* collaboration. Furthermore, they should allow documents to flow seamlessly from synchronous use (e.g., showing slides in a meeting) to asynchronous use (e.g., perusing slides from a past meeting or extracting a photo of a slide so others can refer to it later). Current tools fail to bridge this gap.

We are building a digital project room environment that allows users to collaborate synchronously via web-based meetings as well as asynchronously via a shared document environment. With our system users can annotate and extract content from live meetings both to improve *in situ* communication as well as to allow others to explore important documents later. In this paper we focus in particular on the aspects of our system that allow users to mark-up, extract, and archive live content.

2. INTERACTION DESIGN

2.1 Meeting support

Our system is a pure HTML5[3] solution that utilizes WebRTC[4] to connect clients. It relies on WebRTC media capture and streaming[5] to gain access to a user's webcam and microphone. We use the WebSocket API[6] for communication between the clients and for signalling for the WebRTC connections. Any number of clients may participate in a meeting, limited only by the available network bandwidth and the screen real-estate for displaying content from remote participants. We also utilize the Desktop Capture API[7] to allow participants to stream arbitrary screen regions. Crucially, our system makes no other demands on the end user system. This allows users to share arbitrary documents without forcing other users in a meeting to install third-party applications. Several of those technologies are still under development and are only supported by few web browsers. Still, we expect more mature and widespread support in the near future.

A key feature of our system is that it automatically detects and indexes changes in shared content. When a user changes their content stream, e.g., from their webcam to a screen region, or from one screen region to another, our system records the change event and captures a representative frame of the new content. Furthermore, our system can determine that the webcam has shifted focus from one participant to another via live face detection. This is particularly useful for connecting groups of users who have access to pan-tilt camera systems (e.g., Jarvis[8]). Content frames are archived immediately to an interactive stack that anyone in the meeting can peruse. Also, any user can re-share content from any other users' stack in the same meeting. Captured frames can also be extracted from the meeting and posted to a shared project page.

Figure 1 shows an example meeting with three participants. Stacks for each participant show both the currently shared content (on top of the stack) as well as previously captured frames from other content shared in the meeting. In this case, the two remote users (Tim and Manu) have shared a variety of different content types. The local user (Sue) has only shared her webcam and therefore does not have any archived frames. Note that the local user also has extra controls allowing her to switch her shared content from an arbitrary screen region or her webcam (similar to other web-conferencing tools she can also mute her audio stream). Sue is looking at Tim's stream, which is currently an image of one region of his screen. In Figure 1b,c, we see that Tim has opened Manu's history, has found a slide Manu shared previously, and has annotated it. The annotation snaps to the underlying content which is detected via real-time image analysis. At this point, only Tim sees these actions. However, he can choose to share this popup window (Figure 1d) so that everyone else can see his annotation while he makes a new remark.

In Figure 2, Tim continues to look through Manu's previously shared content and then returns to the slide he annotated. Note that the annotation remains via content matching algorithms we will describe below.

2.2 Archiving meetings

Frames from each user are continuously captured at regular intervals. A keyframe is archived to the user's history whenever a screen change event is detected, while other frames are discarded. Also, users can optionally record their complete streams (both audio and video) during a meeting. In our system, recorded content is collected on the client browser and sent to the server opportunistically. The server maintains a database of all recorded meetings and makes

[3]http://www.w3.org/TR/html5/
[4]http://www.w3.org/TR/webrtc/
[5]http://www.w3.org/TR/mediacapture-streams/
[6]http://www.w3.org/TR/websockets/
[7]http://developer.chrome.com/extensions/desktopCapture
[8]http://www.fxpal.com/research-projects/jarvis/

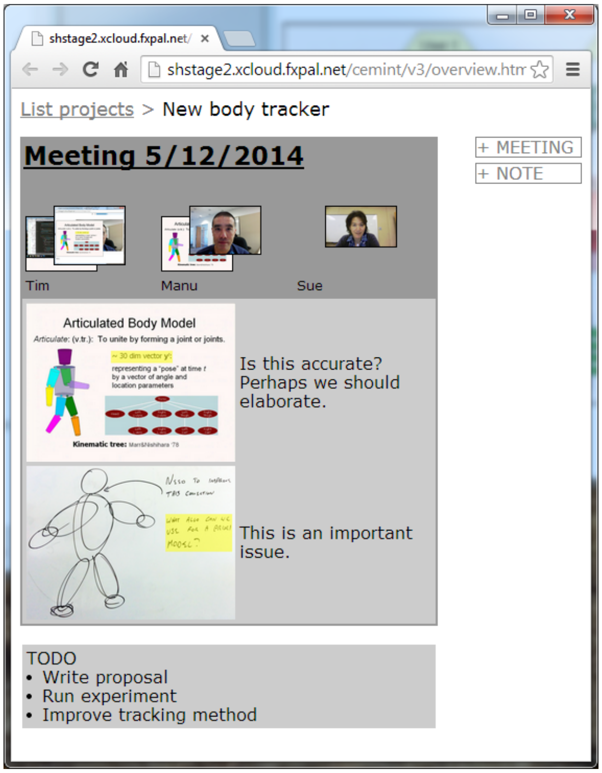


Figure 3: An example project page. Participants can view archived meetings, and notes and media can be attached to archived meeting or directly to the project. Participants can continue an archived meeting or start a new one at any time.

archived content available via a web server.

Users can manipulate a slider (Figure 1a, top) to peruse archived meeting content. As the user moves the slider back, each user's stack is updated to show the archived frame corresponding to that time. During a meeting the ability to peruse past content can be useful both for participants just joining the conference to catch up [5] as well as for ongoing participants who might want to revisit previously discussed topics. When no participants are actively streaming content the user can click a play button to watch the conference. This will play through the video in the main video view. If video streams from multiple participants were recorded, the playback switches among the streams based on speaker identification. Alternatively, the viewer can select one of the participants to keep the focus there.

Note that users can join or rejoin a meeting at any time. Color-coded timelines just below the slider represent the periods of the meeting during which each user was present. If a portion of a client's stream was not recorded, either because they were not in the meeting at that time or because they switched off recording, the user's stack shows their last archived frame.

2.3 Projects

Our goal is for meetings to be integrated into a higher level project page that includes links to a set of related meetings as well as other multimedia and workflow tools. For example, a working group could archive a series of development meetings, design meetings, and HR meetings as well as a shared to-do list on the same page.

We are currently exploring integrating our web meeting tool into open-source organization frameworks (e.g., Redmine[9]). Simultaneously we are building a lightweight project page that allows us to experiment with representing and organizing meetings as well as content extracted from meetings. Figure 3 shows an example project page in which a meeting is represented by the history stacks of each participant. Users can drag arbitrary media into the project page either from their own laptop or from a meeting.

The fact that our annotation approach operates directly on shared screen content, rather than any specific document format, means that users can markup any content, including, for example, photos of sketches. Here, one user has extracted an annotated slide from the meeting, while another has uploaded and annotated a photo of a sketch.

Users can start a new meeting that will be associated with this project by clicking the "+ MEETING" button. They can also add multimedia notes directly to the project page ("TODO..." in the example). Furthermore, users can return to archived meetings at any time to peruse past content or continue the meeting.

3. ENHANCING ANNOTATION USING REAL-TIME ANALYSIS

It is common for one user to annotate the screen content shared by another conference participant. The keyframe history in our interface provides additional opportunities for users to review and optionally markup previously shared content throughout ongoing meetings.

3.1 Creating annotations

Remote participants do not have a way to easily markup content. Previous work on video-based annotation has focused on transient traces or enhanced remote pointers; in other cases markup is not linked to the underlying content and is usually freeform [3]. We utilize real time image processing to detect and segment underlying content in the shared screen. When users elect to annotate (e.g., highlight) underlying content, we integrate the results of the analysis to interpret their interaction.

To segment underlying content we compute a binary version of new frames as they are captured. Content streams such as screens and webcams are rendered into an HTML5 canvas element and we binarize and compute boxes using the canvas pixel data. Connected components [2] are then extracted and their bounding boxes are sorted and grouped to approximate word boundaries. We set a spacing threshold of six pixels for word groups and also filter out boxes that are less than ten pixels wide or high.

Users are shown this canvas element, refreshed in real-time as new video frames arrive. When a user starts a drag operation over the canvas, the system collects the underlying bounding boxes and draws a yellow filled rectangle over the canvas, giving users the impression that they are highlighting the text underneath. Because the rectangle snaps to the connected component boxes, users can create orderly marks as if they were working with an original document, i.e., their highlight follows the content.

[9] http://www.redmine.org/

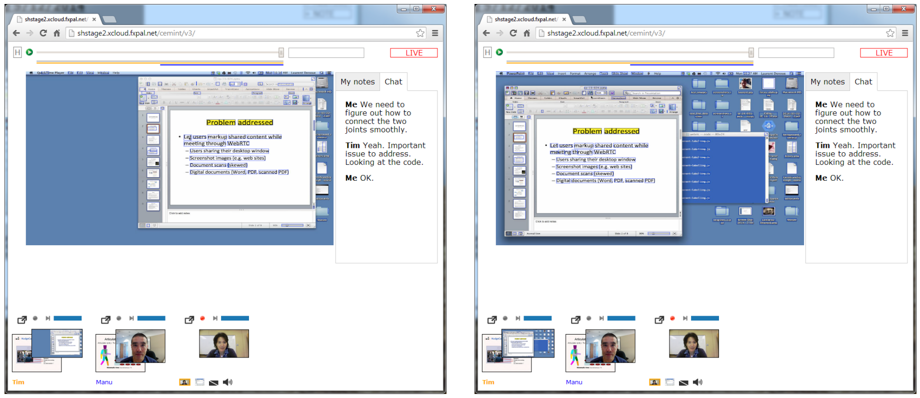


Figure 4: Tracking annotations: word bounding boxes are shown in blue. The user has highlighted 2 words (left). Then the user shifts the window to the left, and the system finds these 2 word boxes in the new frames and repositions the highlight correctly (right).

3.2 Tracking annotations

We are also developing algorithms that allow real-time repositioning of marks created by users. During the natural progression of a meeting, a user may move or scroll a window that has been previously annotated as in Figure 2. The system stores the word bounding boxes that are found under the highlight, and now needs to find where this highlight should be repositioned when new frames arrive.

Our implementation finds a match between the new bounding boxes and the boxes of the highlight based on their dimensions. While this algorithm is simple, we have found it to be fairly robust to the natural translations that occur in meetings. There are a number of extensions to this basic approach that could further improve accuracy such as processing boxes' content, or using spatial context to guide the matching. Document image processing includes a number of methods for word shape analysis that are relevant [6]. When a match is found, the system repositions the highlight(s) at the estimated new location, as shown in Figure 4.

This approach works well when the highlights are long enough (e.g., span more than 3 word boxes). It allowed us to test the usefulness of a system that can reposition marks in real-time: users do not have to worry about moving windows or scrolling documents during a web conference. Marks created by peers are automatically moved and reappear when the same content is shown again, even if at different positions. In a typical scenario, Sue shares her screen to ask feedback about a slide deck she is writing for the team. Team members can highlight parts of the slide and when Sue moves her PowerPoint window on her screen or flips back and forth through slides, any previous marks are automatically redrawn.

4. CONCLUSIONS AND FUTURE WORK

In this work, we introduced a system designed to help remote users conduct and archive web-based meeting content such that it can be organized and extracted to digital project rooms. Our solution is novel in that it 1) automatically and in real time detects and bookmarks different content types from each user's stream (different faces, screens, etc.); and 2) allows users to add annotations linked to content in real time. In this way, users can share and annotate arbitrary content types without forcing others to use any third-party software. Furthermore, because content is archived in real time, users can quickly recover any previously shared content from the current meeting or another archived meeting.

We plan to continue building our web-based conferencing and meeting archive system with particular focus on the design of interfaces for exploring archived meetings. We aim to exploit meetings' persistence in a digital project room to support the types of collaboration that physical project rooms afford. Archived meetings can be reviewed and augmented by additional meeting participants subsequent to the original synchronous meeting, via interaction with the meeting's archived content. We will specifically explore the usefulness of the system's markup and multimedia annotation tools to support this mode of collaboration.

5. REFERENCES

[1] Covi, L. M., Olson, J. S., Rocco, E., Miller, W. J. and Allie, P. A Room of Your Own: What Do We Learn about Support of Teamwork from Assessing Teams in Dedicated Project Rooms? *Proc. of Cooperative Buildings: Integrating Information, Organization and Architecture*. 53–65. 1998.

[2] Chang, F., Chen, C-J. and Lu, C-J. A linear-time component-labeling algorithm using contour tracing technique. *Computer Vision and Image Understanding*. **93**(2):206-220. 2004.

[3] Fagá, Jr., F., Motti, V. G., Cattelan, R. G., Teixeira, C. A. C. and Pimentel, M. da G. C. A social approach to authoring media annotations. *Proc. of the ACM symposium on Document Engineering*. 17–26. 2010.

[4] Gutwin, C. and Penner, R. Improving interpretation of remote gestures with telepointer traces. *Proc. of the ACM conference on Computer Supported cooperative Work*. 49–57. 2002.

[5] Junuzovic, S., Inkpen, K., Hegde, R., Zhang, Z., Tang, J. and Brooks, C. What Did I Miss?: In-meeting Review Using Multimodal Accelerated Instant Replay (Air) Conferencing. *Proc. of the SIGCHI conference on Human Factors in Computing Systems*. 513–522. 2011.

[6] Lu, S., Li, L. and Tan, C.L. Document Image Retrieval through Word Shape Coding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **30**(11):1913–18. 2008.