12 Meeting browsers and meeting assistants

Steve Whittaker, Simon Tucker, and Denis Lalanne

The previous chapter (Chapter 11) explained how user requirements directed our development of meeting support technology, more specifically meeting browsers and assistants. Chapters 3 to 9 discussed the enabling components, i.e. the multimodal signal processing necessary to build meeting support technology. In the following, we will present an overview of the meeting browsers and assistants developed both in AMI and related projects, as well as outside this consortium.

12.1 Introduction

Face-to-face meetings are a key method by which organizations create and share knowledge, and the last 20 years have seen the development of new computational technology to support them.

Early research on meeting support technology focused on group decision support systems (Poole and DeSanctis, 1989), and on shared whiteboards and large displays to promote richer forms of collaboration (Mantei, 1988, Moran *et al.*, 1998, Olson *et al.*, 1992, Whittaker and Schwarz, 1995, Whittaker *et al.*, 1999). There were also attempts at devising methods for evaluating these systems (Olson *et al.*, 1992). Subsequent research was inspired by ubiquitous computing (Streitz *et al.*, 1998, Yu *et al.*, 2000), focusing on direct integration of collaborative computing into existing work practices and artifacts. While much of this prior work has addressed support for real-time collaboration by providing richer interaction resources, another important research area is interaction capture and retrieval.

Interaction capture and retrieval is motivated by the observation that much valuable information exchanged in workplace interactions is never recorded, leading people to forget key decisions or repeat prior discussions. Its aim is to provide computational techniques for analyzing records of interactions, allowing straightforward access to prior critical information. Interaction capture is clearly a difficult problem. A great deal of technology has already been developed to support it (Brotherton *et al.*, 1998, Mantei, 1988, Moran *et al.*, 1997, 1998, Whittaker *et al.*, 1994a), but these systems have yet to be widely used.

Multimodal Signal Processing: Human Interactions in Meetings, ed. Steve Renals, Hervé Bourlard, Jean Carletta, and Andrei Popescu-Belis. Published by Cambridge University Press.

In this chapter, we will consider two main categories of meeting support technology, in relation to the requirements elicited in Chapter 11. We first describe interaction capture and retrieval systems, and then live meeting assistants that have been the focus of more recent research. The first category comprises systems that are designed to enable users to process and understand meeting content, generally after the meeting has taken place. We will present various *meeting browsers*, i.e. user interfaces that support meeting browsing and search, for instance for a person who could not attend a meeting. In contrast, *meeting assistants*, introduced later, are designed to support the real-time meeting process, aiming to increase interaction quality, productivity, or decision making within the meeting itself.

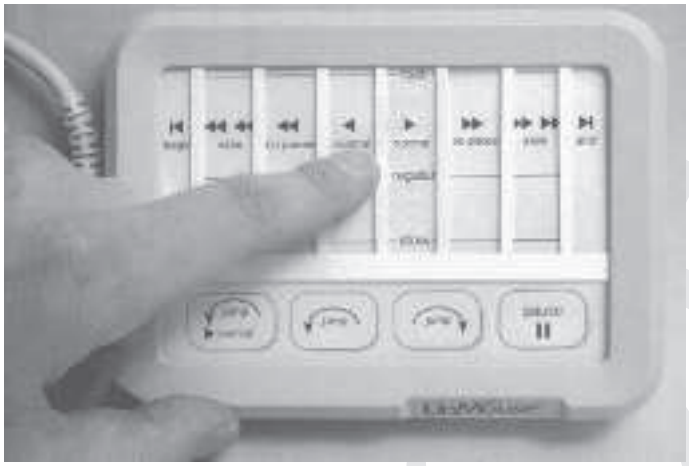
12.2 Meeting browsers

12.2.1 Categorization of meeting browsers

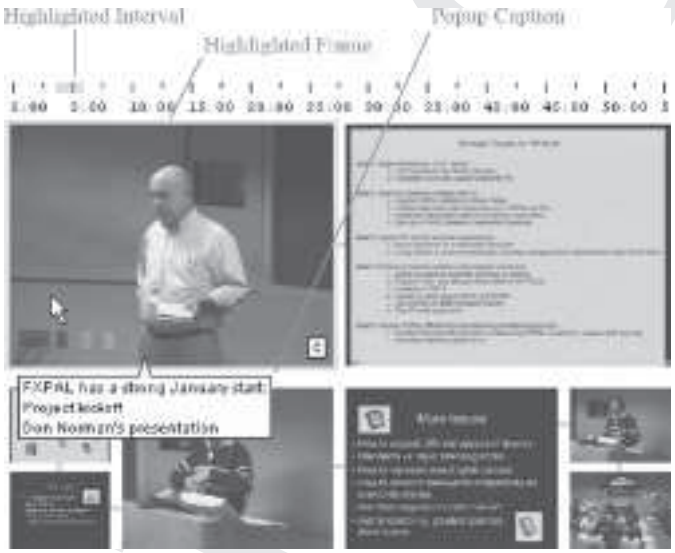
It is possible to categorize different meeting browsers – within interaction capture and retrieval systems – in terms of browser focus (Tucker and Whittaker, 2005). Focus is defined as the main device for navigating the data, or the primary mode of presenting meeting data. We identified four main classes of meeting browsers, shown in Table 12.1. Two classes can be considered as perceptual and two others as semantic, depending on the level of analysis they require.

The first class of browsers focus on audio, including both presentation (Degen *et al.*, 1992, Hindus and Schmandt, 1992) and navigation via audio (Arons, 1997). Others focus on video: examples including video presentation (Girgensohn *et al.*, 2001) or video used for navigation (Christel *et al.*, 1998). The third class of browsers presents meeting artifacts. Meeting artifacts may be notes made during the meeting, slides presented, whiteboard annotations (Cutler *et al.*, 2002) or documents examined in the meeting. All of these can be used for presentation and access. A final class of browser focuses on derived data such as a transcript generated by applying automatic speech recognition (ASR) to a recording of the interaction. Other derived data might include: entities extracted from the recording (names, dates, or decisions), emotions, or speech acts (Lalanne *et al.*, 2003). We call this final class discourse browsers because their focus is on the nature of the interaction.

An example of an audio browser is SpeechSkimmer (Arons, 1997) shown in Figure 12.1(a). Here the device allows the user to browse audio at four different levels of compression – these levels being determined by acoustic properties of the audio source. For example, at the third level only 5 seconds of speech following significant pauses is played back to the user, the significant pause being used here to define a new "unit" of discourse. On top of this acoustic segmentation, the user can alter the playback speed and control the audio stream. This allows the user to quickly navigate to and browse relevant portions of the audio. Figure 12.1(b) shows an example video browser (Boreczky *et al.*, 2000, Girgensohn *et al.*, 2001). These browsers are typically centered around keyframes, static images which are used to represent a portion of the video. The Manga



(a) The SpeechSkimmer Audio Browser



(b) The Manga Video Browser

Fig. 12.1 Audio and video browsers. Reprinted with permission from the publishers, respectively from Arons (1997) and Boreczky *et al.* (2000).

Video Browser shown in Figure 12.1(b) took this further and used the size of keyframes to indicate the relevance of the corresponding video portion. Thus the Manga display is similar to a comic book (similar to SuVi, see Section 8.5.6), drawing the user towards the interesting parts of the video.

Cutler *et al.* (2002) describe a typical artifact browser, shown in Figure 12.2 (a). Although it includes audio and video components, the central focus of the interface is the whiteboard display. The user is able to select annotations made on the whiteboard

(a) An artifact browser focused on a shared whiteboard. Reprinted from Cutler et al. (2002), with permission of the publisher.

(b) FriDoc, a discourse browser which links discourse to documents (Lalanne *et al.*, 2003).

Fig. 12.2 Artifact and discourse browsers.

and navigate to the corresponding point in the meeting. The artifact in question is a community artifact since it can be altered by any of the meeting participants. Figure 12.2 (b) shows FriDoc, a discourse browser developed by Lalanne *et al.* (2003). Here the focus and means of navigation are the speech and interaction that took place in the meeting. In addition, the speech is linked to the relevant documents which were discussed and the interface is time-synchronized so the user is able to use any of the components to navigate around the meeting.

Table 12.1 Main categories of meeting browsers with examples.

Perceptual	Audio	SpeechSkimmer (Arons, 1997)
	Video	Video Manga (Girgensohn *et al.*, 2001)
Semantic	Artifact	Shared Whiteboard (Cutler *et al.*, 2002)
	Derived data	FriDoc (Lalanne *et al.*, 2003)

We refer to audio and video indices as perceptual since they focus on low-level analysis using signal processing methods. Artifacts and derived indices are referred to as semantic since they rely on higher-level analysis of the raw data. Perceptual and semantic systems have different underlying user models. Perceptual systems assume that users will access data by browsing audio or video media selecting regions of interest using random access. In contrast, semantic systems provide higher levels of abstraction, allowing users greater control using search, or by accessing key parts of the meeting (such as decisions and actions). A more detailed taxonomy and review of interaction capture and retrieval systems is provided by Tucker and Whittaker (2005). Given the recent rise of discourse systems that fall within the "Derived data" class in Table 12.1, we discuss some specific examples in detail below.

12.2.2 Meeting browsers from the AMI and IM2 Consortia

The need to address the variability of user requirements, observed in the AMI Consortium and related projects (see Chapter 11), lead to the creation of JFerret, a software platform and framework for browser design. The platform offers a customizable set of plugins or building blocks which can be hierarchically combined into a meeting browser. The platform allows synchronized playback of the signals displayed by the plugins, mainly speech, video, speaker segmentation, and slides. The JFerret framework has been used to implement several browsers, including audio-based, dialogue or document-centric ones, in AMI and related projects (Lalanne *et al.*, 2005b).

A typical instantiation of the platform, often referred to as the *JFerret browser* (Wellner *et al.*, 2006, 2005), is shown in Figure 12.3. This browser is typical of the current state of the art, offering random access to audio and video as well as access via semantic representations such as the speech transcript, and via artifacts such as meeting slides. Audio and video recordings can be accessed directly using player controls. Speech is transcribed, and presented in a transcript containing formatting information showing speaker identification, signaled using color coding for each speaker. The transcript depicted in Figure 12.3 is human-generated and therefore contains no errors, but in general the transcript will be generated using ASR. Clicking on a particular speaker contribution in the transcript begins playing the audio and video related to that contribution. The interface also shows a profile indicating overall contributions of each of the speakers, using the same color coding. This representation can be scrolled and zoomed allowing users to form an impression of overall speaker contribution levels. Finally, the system shows accompanying artifacts including presentations and whiteboard activities. Slides are temporally indexed so that selecting a specific slide accesses other data at

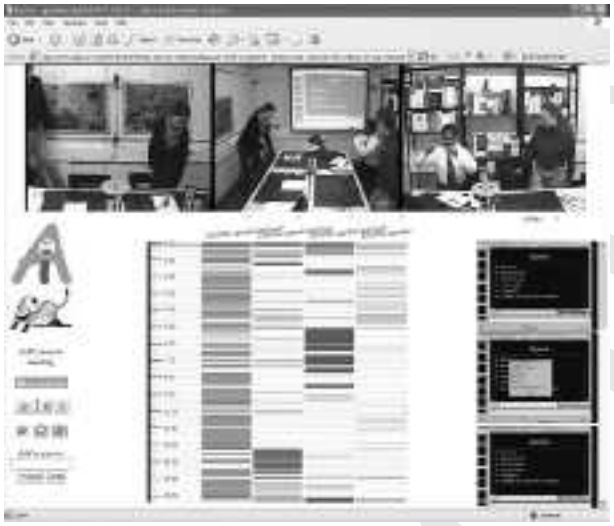


Fig. 12.3 JFerret, a typical meeting browser. Reprinted with permission from Mike Flynn.

that point in the meeting. Whiteboard events are presented as video streams and cannot therefore be used to directly index into the meeting. The JFerret browser has been evaluated by various teams to determine its utility (see e.g., Whittaker *et al.*, 2008, Section 5, and Chapter 13 of this book).

Other browsers have been implemented within the AMI and IM2 Consortia, some focused on audio and speech, and others focused on more media. Three audio-based browsers (AMI Consortium, 2006) were implemented in the JFerret framework (Figure 12.4). They all provide access to audio recordings, with speaker segmentation and slides, and enhance speech browsing in two ways.

The *Speedup browser* accelerates audio playback while keeping speech understandable to avoid the chipmunk effect. Playback is user-controlled allowing 1.5 and 3 times normal playback rates (AMI Consortium, 2006, page 21). The *Speedup browser* includes a timeline, scrollable speaker segmentations, a scrollable slide tray, and headshots with no live video. The speedup method has been extensively user-tested and compared with other methods of speech compression, such as silence removal, unimportant word removal, and unimportant phrase removal (Tucker and Whittaker, 2006). The *Overlap browser* achieves the compression effect in a different way by presenting two different parts of a meeting in the left vs. right audio channels, assuming that the user will take advantage of the cocktail party effect to locate the more relevant channel and then adjust the audio balance to extract the interesting facts (AMI Consortium, 2006, page 22). Again this method was based on extensive experimentation with human subjects to validate the approach and design (Wrigley *et al.*, 2009). Temporal compression of speech was also used in the *Catchup browser*. Catchup allows users to join a meeting late using compression to catch up on the audio content they missed, or more generally to rapidly revisit audio content. As the previous other two, this browser was designed following careful user testing and shown to support comprehension of missed meeting content (Tucker *et al.*, 2008, 2010). Audio-based browsers require very little

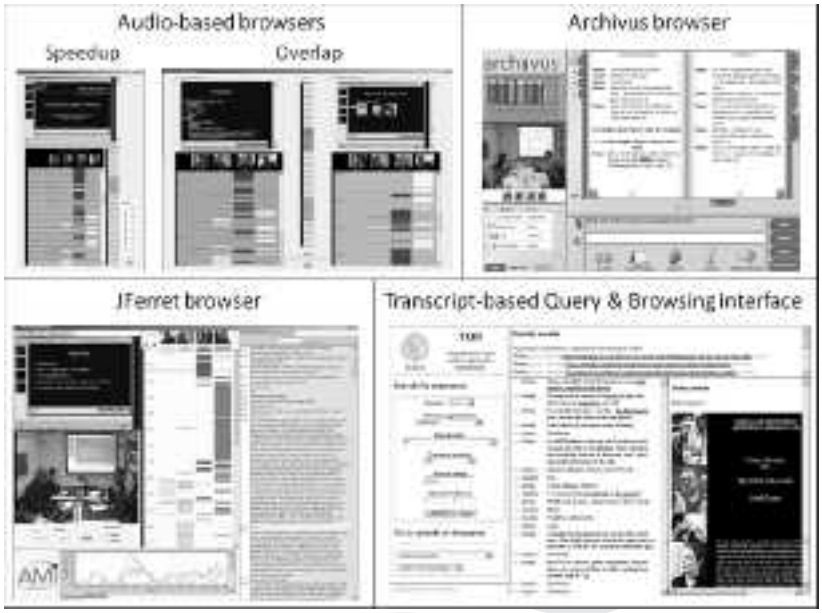


Fig. 12.4 Five speech-centric meeting browsers from the AMI and IM2 consortia, illustrating the diversity of media and layouts. Components include audio, video, and slide players, along with speaker identification and segmentation, transcript, and various query parameters in Archivus and TQB. Reprinted with permission from Agnes Lisowska-Masson (Archivus) and Mike Flynn (JFerret).

human preparation of automatically recorded data before use, and their performance on information extraction tasks as well as summarization is clearly encouraging (see Chapter 13, Section 13.3.2 on browser evaluations).

Several other browsers implemented within the AMI and IM2 Consortia were focused on more media than speech. In addition to the JFerret framework and browser mentioned above, the *Transcript-based Query and Browsing (TQB) interface* (Popescu-Belis and Georgescul, 2006, Popescu-Belis *et al.*, 2008a) is another speech-centric browser, which provides a number of manual (reference) annotations in order to test their utility for meeting browsing: manual transcript, dialogue acts, topic labels, and references to documents. These parameters can be used to formulate queries to a database of meeting recordings, and have been tested with human subjects on the BET task (see again Chapter 13, Section 13.3.2). The evaluation results are also used to set priorities for research on the automatic annotation of these parameters on meeting data.

Archivus (Ailomaa *et al.*, 2006, Melichar, 2008) is a partially implemented meeting browser that supports multimodal human–computer dialogue. Its purpose was to gather user requirements (Lisowska *et al.*, 2007), especially with respect to modality choice, using a Wizard-of-Oz approach. Archivus uses reference transcripts enriched with annotations (speaker segmentation, topic labels, documents) to answer user queries that are expressed as a set of attribute/value constraints over one or several meetings. An implementation using a standalone dialogue engine with a multilingual front-end and a touch-screen on a mobile device was built for a subset of the Archivus search attributes, as the *Multilingual Multimodal Meeting Calendar (M3C)* (Tsourakis *et al.*, 2008).

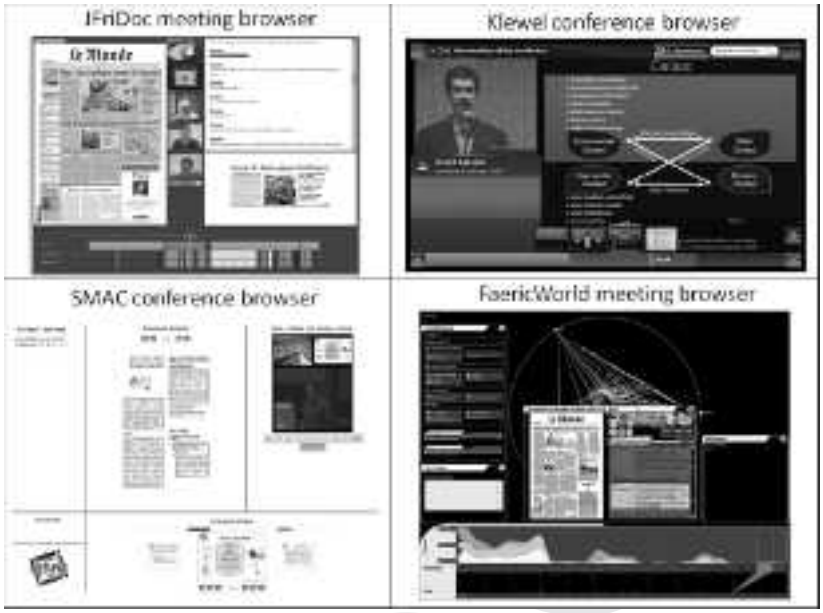


Fig. 12.5 Document-centric meeting browsers and conference browsers from the AMI and IM2 Consortia described in the text. Document/speech alignment is central to all layouts. Reprinted with permission from Maël Guillemot (Klewel).

FriDoc (Lalanne *et al.*, 2005a) and *JFriDoc* (Rigamonti *et al.*, 2006), are document-centric browsers that link documents discussed during a meeting, dialogue transcripts, slides, and audio-video streams. They exploit automatic alignments between printed documents and speech as well as video (see Figure 12.5), highlighting when a document section was discussed during a meeting (by automatic alignment of document content with speech transcript content), or when a document was the visual focus (by automatic alignment of document image with video of projection screen, or document on the table). In these browsers, clicking on a specific document part (e.g. a section, an image, etc.) accesses the audio/video recording at the moment when the content of that document section is being discussed. In the same way, selecting a moment in the audio/video stream will automatically select the relevant document section. The benefit of this automatic alignment has been evaluated, and proven to be useful for meeting browsing, using the methods described in Chapter 13.

Similarly, *ViCoDe* (*Video Content Description and Exploration*) computes the similarity between speech and document sentences. When combined with relevance feedback, this supports new ways of browsing meetings (Marchand-Maillet and Bruno, 2005). *FaericWorld* (Rigamonti *et al.*, 2007) enhances document-based browsing with cross-meeting representations of documents and links. For each collection of meetings, links between all multimedia data associated with the meetings are automatically derived through an analysis of the input streams upon indexing of the meeting into the system's database. Users can then query the system with full text search or directly browse through links, using interactive visualizations. Finally, *WotanEye* (Évequoz and

Lalanne, 2009) enables ego-centric access to meeting fragments using personal cues, such as the user's social network.

An extension of the discourse browsing approach includes the analysis and presentation of an entire meeting through some form of summarization, for instance as presented in Chapter 10. Variants on this include analyzing the meeting to identify important discourse acts, allowing users to focus directly on decisions or on items to do (Fernández *et al.*, 2008). Another approach has been exemplified by the Summary Visualizer (SuVi, see Section 8.5.6), which uses the automatic extractive or abstractive summaries based on ASR, together with video information, to create a multimodal storyboard (or comic book) meeting summary (Castronovo *et al.*, 2008). The output can be visualized and printed, but can also be used in HTML format within a more complex meeting browser.

12.2.3 Conference recording and browsing

Despite the large number of research prototypes, there are still no commercially available end-user meeting browsers. This is all the more surprising since some of the commercially available systems for coordinating remote meetings offer recording capabilities, but no support for more advanced browsing (other than replay). The meeting browsers developed within AMI and related projects have evolved towards two end-user products, but for a slightly different task, namely conference recording and browsing. The two products answer a growing need for conference recording in flexible settings and playback using cross-platform, user-friendly interfaces, as initiated for instance in the Classroom 2000 educational environment (Abowd, 1999). These two applications to conference recording and browsing use fewer capture devices than instrumented meeting rooms, and use off-the-shelf technology rather than capture devices designed on purpose, resulting in smaller amounts of data to store and process, which might explain why they were quicker to reach product stage.

One system is commercialized through a spin-off company of the Idiap Research Institute named Klewel (www.klewel.com), while the other one was developed by the University of Fribourg and the CERN in Geneva within the SMAC project (Smart Multimedia Archive for Conferences, http://smac.hefr.ch) and is in use at these institutions. Both systems extract a number of robust indexes, such as slide changes, text from slides, and slide/audio/video synchronization, which are helpful for browsing, and provide some support for fact-finding. The SMAC system, in addition, is able to automatically hyperlink the fragments of the scientific article that is being presented to the related audio-video sequence (Lalanne *et al.*, 2004). Such technologies derived from our consortia give these browsers an advantage over other competing systems (Herr *et al.*, 2010).

12.3 Meeting assistants: real-time meeting support

To demonstrate how component technologies might be combined to address some of the user requirements presented in the previous chapter, several other applications have

been designed and implemented by members of the AMI Consortium or related projects. Although the initial focus was on meeting browsers, it shifted toward real-time meeting assistants that aim to increase the efficiency of an ongoing meeting. The achievements thus cover the multiple facets of meeting support addressing user needs before, during, and after a meeting.

Several pieces of software infrastructure were designed to support the implementation of demonstrators. The Hub is a subscription-based client/server mechanism for real-time annotation exchange (AMI Consortium, 2007). The Hub allows the connection of heterogeneous software modules, which may operate remotely, ensuring that data exchange is extremely fast – a requirement for real-time meeting support. Data circulating through the Hub is formatted as timed triples (time, object, attribute, value), and is also stored in a special-purpose database, which was designed to deal with large-scale, real-time annotations and metadata of audio and video recordings. "Producers" of annotations send triples to the Hub, which are received by the "consumers" that subscribe to the respective types; consumers can also query the Hub for past annotations and metadata about meetings. The HMI Media Server (see op den Akker *et al.*, 2009) complements the Hub for media exchange. It can broadcast audio and video captured in an instrumented meeting room to various "consumers," thus allowing a flexible design of interfaces that combine the rendering of media streams with annotations and metadata. The server is built on low-level DirectShow filters under Microsoft Windows, thus providing accessible interfaces in C++ and Java, and can stream media over UDP network ports to multiple targets.

12.3.1 Improving user engagement in meetings

An important requirement for meeting assistants is to improve the meeting experience for participants attending remotely. The objective is to go beyond simply exchanging audio and video between remote participant(s) and physically co-located ones. AMI processing technologies can be used to enrich the audio and video with information to help remote participant(s) to better understand the communication going on within the meeting, allowing them to intervene more efficiently in the discussion. Two such meeting support applications were designed by AMI and IM2 Consortia members: one intended for users connected through a mobile device, and the other one for users connected through a desktop or laptop computer.

The Mobile Meeting Assistant (MMA) is a prototype mobile interface aimed at improving remote access to meetings (Matena *et al.*, 2008). Remote participants often complain that they have little idea about the underlying interpersonal dynamics of meetings (e.g. gestures or eye gaze), and providing high-quality video data is still not possible with today's mobile devices. Unlike more traditional teleconferencing devices, the MMA allows remote users not only to hear other participants and to view projected material (slides), but also to gain insights into their nonverbal communication. Two main modes were designed to display a representation of the physically collocated group on the remote participant's mobile device: a two-dimensional (2D) and a three-dimensional (3D) representation, both shown in Figure 12.6.

Fig. 12.6 The 2D and 3D interfaces of the Mobile Meeting Assistant (Matena *et al.*, 2008).

The MMA prototype uses graphical elements to represent nonverbal information related to the audio-visual behaviors of the co-located participants, including: (1) speaking status, inferred from ASR and speaker segmentation (see Chapters 4 and 5), shown by red moving lips; (2) head orientation obtained through video processing (see Chapter 6); and (3) individual or joint visual focus of attention obtained through multimodal processing (see Chapter 6 and Chapter 9, Section 9.3.1), represented in the 3D view by a green arrow. A user evaluation was performed using a meeting from the AMI Corpus (see Chapter 2) with 13 subjects who acted as remote participants (see for details Matena *et al.*, 2008, AMI Consortium, 2008). Feedback from these subjects, as well as from industrial partners in the AMI Community of Interest, was overall positive. It appeared however that the graphical conventions should be improved, and more information about the participants should be provided.

The User Engagement and Floor Control (UEFC) prototype trades mobility for higher computing power, bandwidth, and size of display (op den Akker *et al.*, 2009). The UEFC is motivated by the fact that, in meetings, remote participants are often multi-tasking (e.g. reading email while listening to the ongoing meeting conversation), and might benefit from receiving alerts when specific keywords are uttered, or when they are addressed by one of the co-located group's members. The UEFC integrates keyword spotting to support alerts for selected keywords (see Chapter 5, Section 5.6), along with visual focus of attention and online addressee detection, which provide alerts about when the remote participant's image becomes the focus of attention of local participants. The interface of the UEFC system is shown in Figure 12.7. The dedicated addressee detector uses lexical features from the ASR, and the output of the visual focus of attention analyzer (see Chapter 9, Section 9.3.1), for a binary decision task (whether the remote participant is being addressed or not). The online dialogue act segmentation and labeling (see Chapter 8, Section 8.2) are also integrated.

12.3.2 Suggesting relevant documents during meetings

Participants in meetings often need access to project-related materials (e.g. meeting minutes, presentations, contracts, specification documents) but they often do not have

Fig. 12.7 The User Engagement and Floor Control System (op den Akker *et al.*, 2009). Reprinted with permission from Rieks op den Akker.

the time during the meeting to search for these. Similarly, they may want to access recordings of their past meetings, but again do not want to disrupt a meeting to search for them. The Automatic Content Linking Device (ACLD, see Popescu-Belis *et al.*, 2008b, 2011b) is a meeting support application that provides just-in-time and query-free access (as in Hart and Graham, 1997, Rhodes and Maes, 2000) to potentially relevant documents or fragments of recorded meetings. The ACLD thus provides automatic real-time access to a group's history, presented as suggestions during an ongoing meeting.

The ACLD makes use of speech-oriented AMI core technologies such as automatic speech recognition and keyword spotting (see Chapter 5) and speaker diarization (Chapter 4), using the Hub to exchange annotation and the HMI Media Server to broadcast media. The main ACLD component is the Query Aggregator, which performs document searches at regular time intervals over a database of previous documents and meeting transcripts (e.g. from the AMI Corpus, see Chapter 2), using words and terms that were recognized automatically from the meeting discussion. While the first prototypes used Apache Lucene for keyword-based search in local repositories, a more recent version uses "semantic search" to cope with noise in ASR and to improve the relevance of search results (Popescu-Belis *et al.*, 2011b). The Query Aggregator is also connected to the Google search engine, and separately manages a list of the top hits retrieved from a user-specified web domain.

The ACLD output shown to users is a list of document names ordered by relevance, refreshed at regular intervals (15 seconds) or on demand, based on the search results and on a persistence model which ensures that documents that are often retrieved persist at the top of the list. The snapshot in Figure 12.8 shows the user interface of the ACLD in a detailed view, with all four widgets visible: ASR words, tag cloud of keywords, document results (with pop-up window open when hovering over a name), and Web results. An unobtrusive view can display the widgets as superposed tabs, freeing

Fig. 12.8 User interface of the Automatic Content Linking Device (Popescu-Belis *et al.*, 2011b).

up screen real-estate for other activities. Evaluation results for the ACLD have shown that users clicked on a suggested document every 5–10 minutes, that they found the UI "acceptably" usable, and that results of semantic search were found five times more relevant than those of keyword-based search.

12.4 Summary and perspectives

This chapter presented two types of meeting support technologies answering some of the most important requirements that were found by the AMI Consortium and other projects. The first type (meeting browsers) supports capture, post hoc analysis, and replay of meetings, whereas the second one (meeting assistants) is used during meetings to enrich live interactions between meeting participants. Several meeting browsers have been described, making use of raw video and audio recordings, of artifacts such as whiteboard recordings or documents projected or discussed during the meeting, or using annotations derived from raw data recordings, such as the speech transcript or the visual focus of attention.

Despite the number of research prototypes for meeting browsing, none of them have achieved large-scale mass adoption. One reason for this lack of uptake is socio-technical issues that have to be addressed before systems become acceptable. For instance, in various user studies (e.g. starting with Whittaker *et al.*, 1994a), users expressed concerns about privacy, and about the impact of being recorded on the process of the meeting itself. This is possibly one of the reasons why, from the numerous browsers developed by AMI and related projects, the two resulting end-user products are those aimed at the recording and browsing of public conferences.

There are a number of important practical and research issues arising. For meeting browsers the technology is relatively well understood, but two main areas remain

to be addressed. The first concerns data capture: basic approaches to recording high-quality multimedia data are not standardized, with most meeting rooms currently lacking recording equipment. Without such data we cannot build successful browsers. The second issue relates to user value: meeting participants seem remarkably resistant to changing meeting practices, and in many studies have not embraced the opportunity to re-access recordings of past meetings (see e.g. Whittaker *et al.*, 2008). We need a better understanding of why this is the case, as well as an understanding of the situations and contexts in which participants would value such access.

Turning to real-time assistants, here the field is much more open to developing new types of tools based on analyses of ongoing behavior. Such analyses might extend to complex dialogue issues such as conflict and debate, which might improve fundamental meeting processes. New systems might identify if particular participants are dominating a discussion or whether a discussion is leading to an unresolvable impasse. They might detect when there are implicit disagreements or help participants better understand their common ground. Again however the history of prior work has shown that meeting interactions are highly sensitive to disruption so any new technology must be designed to integrate well with existing meeting practices.