The European Commission
Community Research
Information Society Technologies

m4
MULTIMODAL
MEETING MANAGER

Deliverable D4.3:

Report on Final Demonstrator and Evaluation

25 February 2005

Project ref. no.	IST-2001-34485
Project acronym	M4
Project full title	MultiModal Meeting Manager
Security	Public
Contractual date of delivery	M36 (end March 2005)
Actual date of delivery	31 March 2005
Deliverable number	D4.3
Deliverable name	Report on Final Demonstrator and Evaluation
Type	Report
Status and version	Final v1.0
Number of pages	31
WP contributing to deliverable	WP4
WP/Task responsible	IDIAP
Other contributors	USFD, EPFL, TUM, TNO, UniGe, VUT Brno
Editor(s)	Pierre Wellner, Mike Flynn
EC project officer	Domenico Perrotta
Key words	Information access, multimodal information
Abstract	This document describes the M4 Multimodal Meeting Archive Browser application, and its evaluation. The application allows browsing of recorded multi-modal meetings through user-selected visualizations of multiple different segmentations including speaker identification, automatic speech recognition, automatically recognized meeting actions, slides, whiteboard strokes, and textual transcripts.

Contents

1 Introduction

This report presents the technical documentation of the Ferret Media Browser, and a description of how it was evaluated. The goal of Ferret is to bring together the diverse technologies developed in the other projects to create integrated multimodal offline meeting management tools. It will ultimately be the goal of this WP to address the portability and standardization issues, as well as the final evaluation of the system.

As fully described in [10], the IDIAP smart meeting room is in many ways an ordinary conference room with a table, whiteboard, and computer projection screen. But the room is also equipped with 24 microphones configured as lapel microphones, in the ears of a binaural manikin, and in two 8-channel tabletop microphone arrays (See Figure 1). The experimental smart meeting room is set up such that it records multi-party meetings with dozens of multimodal data streams and then processes these media streams to recognize events of potential interest. Effective browsing of recorded meetings requires the careful combination of both automatic recognition as well as interactive browsing.

In this report, we describe how multi-modal meeting recordings can be displayed for interactive browsing. The approach makes use of processing techniques [17] that recognize structure within these meetings, but combines automatic processing with manual interactive exploration of multimedia representations of the recordings.

The remainder of this report is structured as follows: the following section (§2) describes the current state of the art in meeting browsing; the next section (§3) discusses the goals of the Ferret browser, followed by a description of the type of data available to browsers (§4). A user's view of the browser is presented in (§5). A technical overview of the Ferret architecture and principal components is then presented (§6).

Further detailed software documentation is referenced where applicable.

2 Background

This section presents an overview of existing systems and recent contributions in the field of meeting browsing.

With current audio-visual archives, image and video collections, tools that enable fast browsing are fundamental. We inherit from a large literature in the field of video – TV news, sport, home consumer videos – (for example, the Físchlár system from Dublin University [18], or CMU's Informedia [19]) and still image browsing systems. Also the work from Rogina et al. [8] and projects such as the meeting room capturing system at ICSI, and ScanMail (from AT&T) concentrated on audio indexing and browsing.

Most systems are based on a unique modality, and also need interaction analysis to put the user in the loop. Two recent browsers [1] from Microsoft research and [2] from Ricoh use both audio and video recordings from a meeting room. The contributions [4, 5, 6] describe the FXPAL smart conference room and their work on semi-automatic meeting minutes. An Informedia project team at CMU studied on understanding why capturing and accessing meetings is also valuable for teamwork [3]. More generally, Rui et al [7] applied multimodal analysis to teleconferencing, tele-presentation and distance learning.

The Ferret system that is presented here is built upon Stephane Marchand-Maillet's SVG-based browser [14] and Pierre Wellner's MMM browser (http://mmm.idiap.ch).

3 Aims of the Ferret browser

The aim of the Ferret browser is to demonstrate M4 project outputs such as media of recordings, segmentations, transcripts and ASR results. From an HCI point of view, Ferret must offer a service to navigate interactively within meeting recordings, and quickly find and play back segments of interest within these recordings

The current version of the application allows browsing of recorded multi-modal meetings through user-selected visualizations of multiple different segmentations: speaker identification, automatically recognized meeting actions, and textual transcripts.

In this section, current data streams generated after the multi-media signal analysis phase are described. All recorded data is precisely synchronized so that every microphone, pen strokes, and video sample can be associated with simultaneously captured samples from other media streams.

4 Ferret data sources

4.1 Overview

Ferret can display data from many different sources, simultaneously. These include basic media recordings, processed media recordings, and further derived data such as segmentations, transcripts, and so on.

In order to make the data manageable, Ferret directly supports only three fundamental data-types:

- video and audio recordings, using RealMedia formats;
- segmentation data, using a simple XML data format;
- textual data (either manual or ASR-generated transcripts), as HTML files.

However, Ferret can also use:

- slide data, converted to HTML, with embedded images;
- whiteboard and pen recordings, converted to videos;

Of course, any other data that can be converted into the above-mentioned formats can be used directly.

The following sub-sections describe the kind of data available in the research communities, which demanded such support.

4.2 Speaker localisation and segmentation

The approach being performed, described in [12], by Lathoud et al. makes use of tabletop microphone arrays that are capable of tracking the directions which speech is coming from around the table, and it can detect which participants are speaking at any time. This output is used in the browser to show when a participant speaks, making it possible for the user to see aspects of global meeting structure such as when a particular person made a presentation or when a discussion occurred.

4.3 Group action recognition

It is possible to detect some of these meeting actions previously enumerated automatically. In experiments described in [9] McCowan et al, the audio and visual features for each participant are extracted from the raw data, and the interaction of participants is modelled using Hidden Markov Model based approaches. Testing these models on an initial corpus recorded at IDIAP demonstrated the ability of the system to recognize a set of meeting actions, including presentations, monologues, and discussion.

Such an approach poses several interesting challenges for machine learning, related to the modelling of multiple, interactive, asynchronous streams of multi-modal data.

4.4 Level of interest

Current research is performed to try to recognize the level of interest during a meeting. One goal of the ferret browser is to assist researchers to visualize results, and conducting user studies on a particular aspect of research.

4.5 Other recognition research

Researchers investigate the following areas of interest for pattern recognition:

- *Speaker turn and floor detection, contribution at individual level*: Speaker localization, segmentation. Detection of who has the floor.
- *ASR, Dialog acts, activity recognition*: Large vocabulary speech recognition, and more: dialog acts (agreement, etc.) recognition, activity (brainstorming, argumentative monologue, alternative design, etc.) recognition.
- *Audio-visual tracking*: Recent work combining both computer vision and speech analysis fields lead to systems enabling audio visual speaker tracking [13]. This is especially useful for detecting who is speaking at a given time and to detect participant faces before recognition.
- *Level Of activity, Interactivity*: Research is performed in recognizing people activity as well as measuring level of interactivity.
- *Head pose estimation, eye gaze, focus of attention (individual and group)*: For person identification and face verification, the face is not always perfectly positioned. Head pose estimation is useful for face verification and also for eye gaze detection.
- *Gesture recognition*: Also in current research, recognizing people's motion and behaviour may help in recognizing aspects of meetings.
- *Facial expression recognition*: Recognizing patterns of emotion and personal states is also being investigated for integration in our experimental browser.

More details of this research is available in [17].

5 The Ferret browser

5.1 Overview

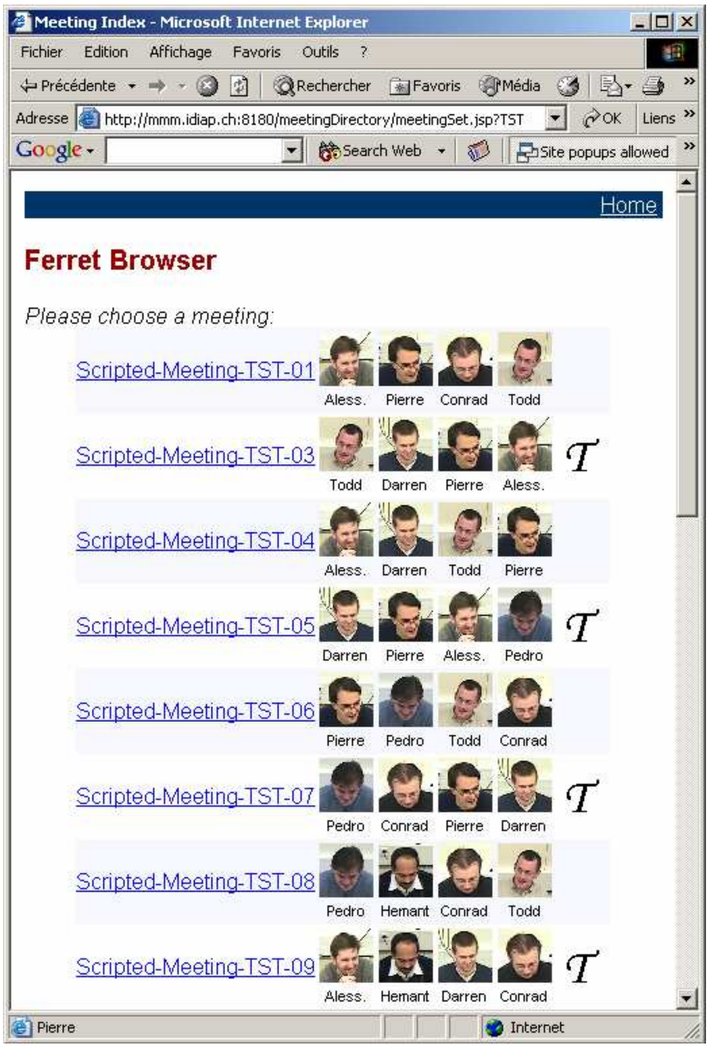
The Ferret browser is typically used in four stages:

- *Meeting selection*, where one meeting is chosen from many;
- *Initial data choice*, where the data to view is chosen;
- *The Ferret browser* itself, where the meeting is browsed;
- *Addition and removal of data.*

These are discussed in the following sub-sections.

5.2 Meeting selection

The user first chooses a meeting. Figure 1 below illustrates an example of a browser to navigate within a set of meetings with information on which people participated at a particular meeting, as well as selected data available to browse.


Meeting Index - Microsoft Internet Explorer
Fichier Edition Affichage Favoris Outils ?
Précédente Rechercher Favoris Média
Adresse http://mmm.idiap.ch:8180/meetingDirectory/meetingSet.jsp?TST OK Liens
Google Search Web Site popups allowed
Home
Pierre
Internet

Ferret Browser

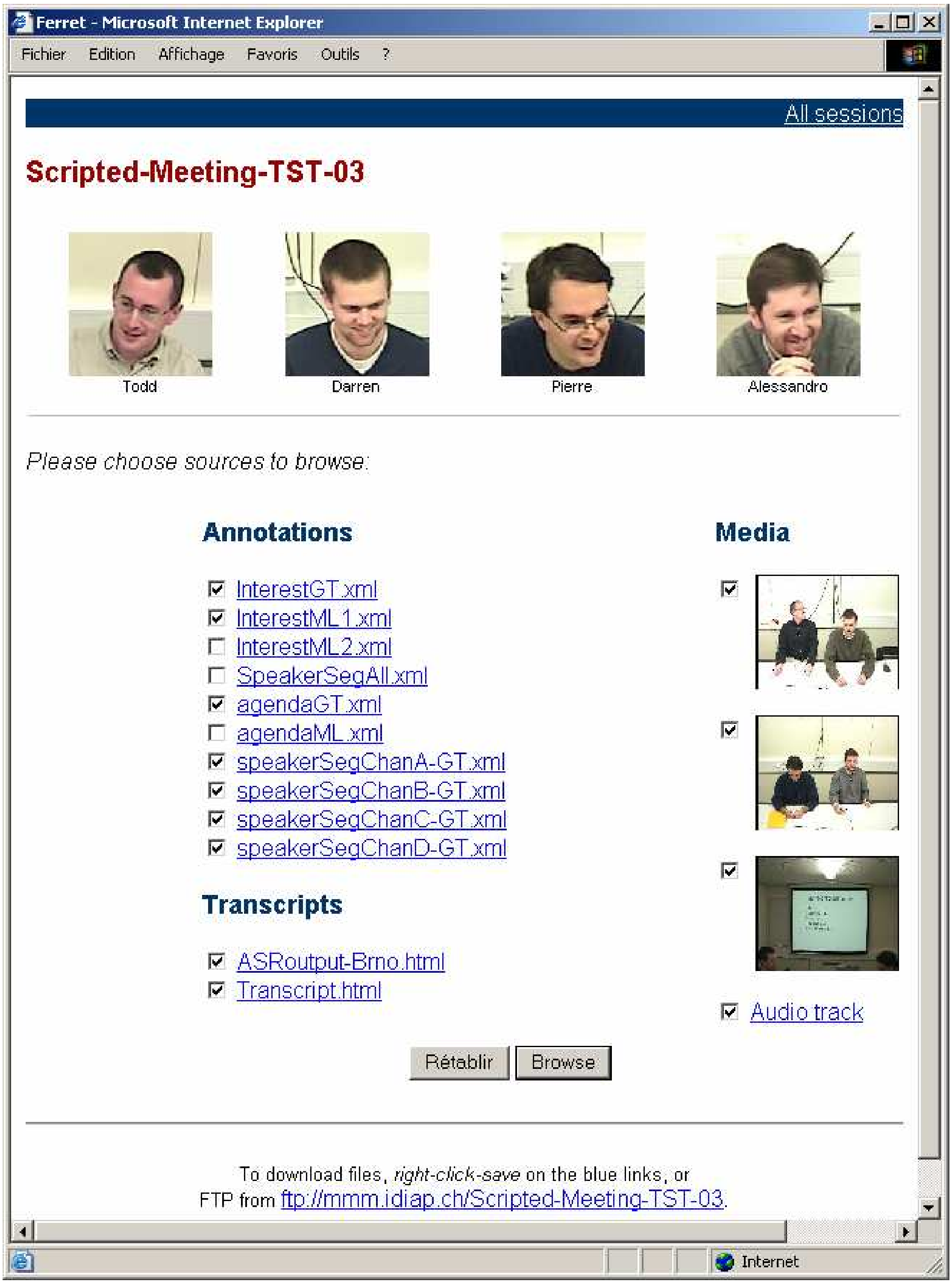
Please choose a meeting:

Scripted-Meeting-TST-01	Aless.	Pierre	Conrad	Todd	
Scripted-Meeting-TST-03	Todd	Darren	Pierre	Aless.	T
Scripted-Meeting-TST-04	Aless.	Darren	Todd	Pierre	
Scripted-Meeting-TST-05	Darren	Pierre	Aless.	Pedro	T
Scripted-Meeting-TST-06	Pierre	Pedro	Todd	Conrad	
Scripted-Meeting-TST-07	Pedro	Conrad	Pierre	Darren	T
Scripted-Meeting-TST-08	Pedro	Hemant	Conrad	Todd	
Scripted-Meeting-TST-09	Aless.	Hemant	Darren	Conrad	T

Figure 1 – Inter-meeting browser prototype

5.3 *Initial data choice*

The user has to select what particular features are of interest. Figure 2 below shows an example for one meeting.



Scripted-Meeting-TST-03

Please choose sources to browse:

Annotations

- ☑ InterestGT.xml
- ☑ InterestML1.xml
- □ InterestML2.xml
- □ SpeakerSegAll.xml
- ☑ agendaGT.xml
- □ agendaML.xml
- ☑ speakerSegChanA-GT.xml
- ☑ speakerSegChanB-GT.xml
- ☑ speakerSegChanC-GT.xml
- ☑ speakerSegChanD-GT.xml

Transcripts

- ☑ ASRoutput-Brno.html
- ☑ Transcript.html

Media

- ☑
- ☑
- ☑
- ☑ Audio track

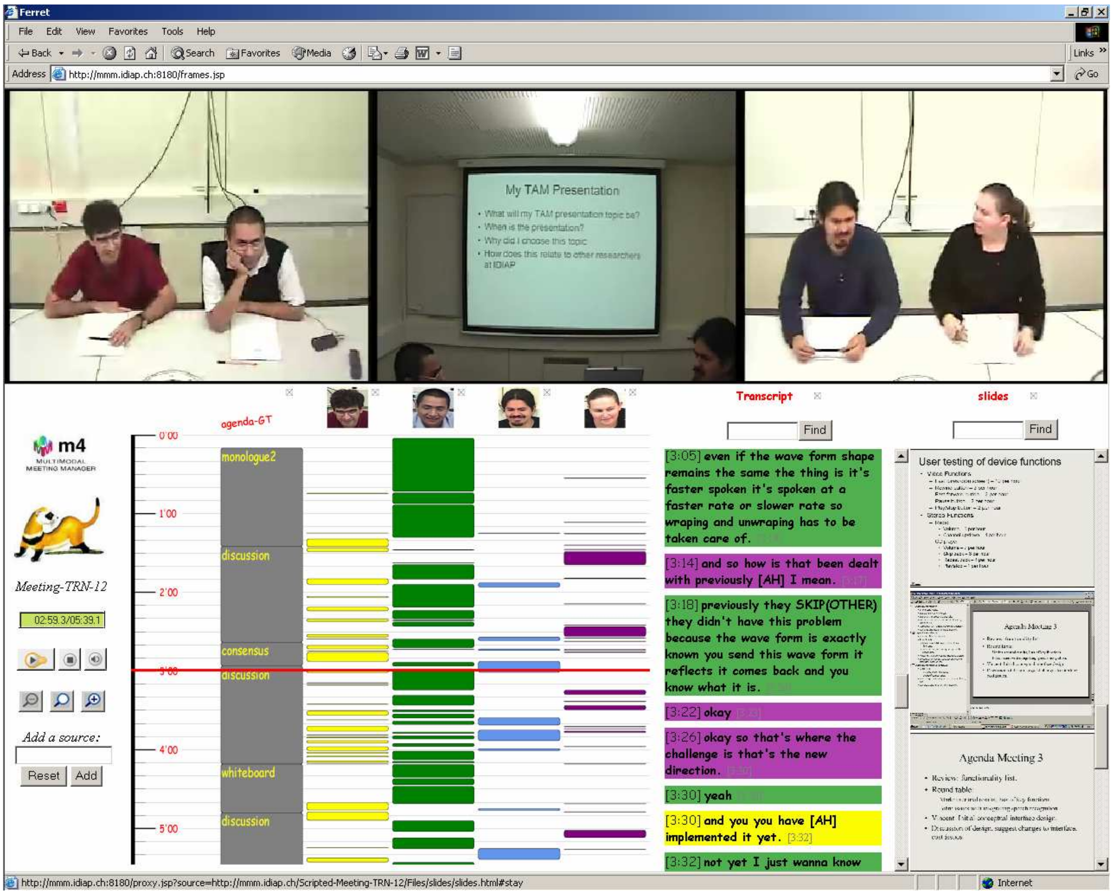
To download files, *right-click-save* on the blue links, or
FTP from ftp://mmm.idiap.ch/Scripted-Meeting-TST-03.

Figure 2 – Meeting data page

This JSP page displays the available annotation files and media files. Once a user has selected which XML streams, HTML transcripts / ASR output and other medias that they need, pushing the "Browse" button brings the user to the meeting browser.

5.4 The Ferret browser

The upper part of the graphical user interface contains a media player for synchronized playback of audio and video. This allows viewing and hearing what happened at particular moment of the meeting as with a multi-TV system, and it also can display stokes written on the whiteboard. The VCR controls (play, pause, sound adjustment, time display) are on the left pane. This left frame controls also the interval data streams.


Ferret
File Edit View Favorites Tools Help
Back Search Favorites Media
Links
Address http://mmm.idiap.ch:8180/frames.jsp
Go
My TAM Presentation
What will my TAM presentation topic be?
When is the presentation?
Why did I choose this topic
How does this relate to other researchers at IDIAP
m4
MULTIMODAL MEETING MANAGER
Meeting-TRN-12
02:59.3/05:39.1
Add a source:
Reset
Add
agenda-GT
0'00
1'00
2'00
3'00
4'00
5'00
monologue2
discussion
consensus
discussion
whiteboard
discussion
Transcript
Find
slides
Find
User testing of device functions
Video Functions
Stereo Functions
Agenda Meeting 3
Review: functionality list.
Round table
Discussion of design, suggest changes to interface.
http://mmm.idiap.ch:8180/proxy.jsp?source=http://mmm.idiap.ch/Scripted-Meeting-TRN-12/Files/slides/slides.html#stay
Internet

[3:05] even if the wave form shape remains the same the thing is it's faster spoken it's spoken at a faster rate or slower rate so wraping and unwraping has to be taken care of.

[3:14] and so how is that been dealt with previously [AH] I mean.

[3:18] previously they SKIP(OTHER) they didn't have this problem because the wave form is exactly known you send this wave form it reflects it comes back and you know what it is.

[3:22] okay

[3:26] okay so that's where the challenge is that's the new direction.

[3:30] yeah

[3:30] and you you have [AH] implemented it yet. [3:32]

[3:32] not yet I just wanna know

Figure 3 - The Ferret browser

A large number of processed interval data streams can be made available to assist with browsing of the meetings. Ferret allows users to select any combination of available data streams as described in the previous section and display them alongside each other for inspection and comparison (see Figure 3 above). Interval data stream graphical representations are displayed on a timeline in the lower part of the browser. These can be speaker turn intervals, meeting actions, level of interest representation, textual transcript as described in the previous section. Clicking on elements in the timeline (vs. textual transcript) controls playback in the media player as well as the scrolling of the textual transcript (vs. timeline).

The user can zoom in particular parts of interest by means of the zooming button on the left. Also by zooming out, the user gets an overview of the meeting in terms of who talked a lot, what meeting actions etc. Crosses at the top of each stream allow for deleting streams. The Add button in the control panel permits to add a new XML or HTML annotation stream to be visualized.

5.5 Addition and removal of data

In addition to the data presented initially, a user may add data of their own choosing to the display. This data does not need to be on any special server – but it must be accessible from the Internet.

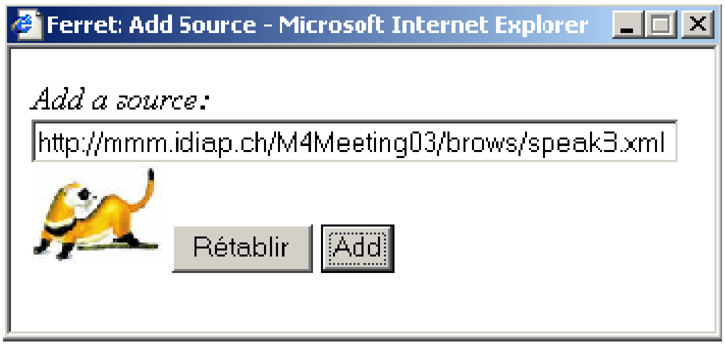

Ferret: Add Source - Microsoft Internet Explorer
Add a source:
http://mmm.idiap.ch/M4Meeting03/brows/speak3.xml
Rétablir
Add

Figure 4 – The Add Source dialog box

In order to display it, the user presses the “Add Source” button, and a dialog appears, illustrated in Figure 4 above. Entering the URL of the new data, and pressing the Add button, the data is displayed alongside the previously selected data[1]. In effect, the Ferret browser is a kind of “shopping basket” into which many disparate sources of data may be placed.

Data can be removed from the main display by pressing the small close button associated with each source.

6 Ferret architecture

6.1 Overview

Ferret works over the Internet, using a normal web browser as the client application. The main meeting corpus also resides on the Internet, along with much processed data – such as transcripts and recognition results.

[1] At the moment, only new XML and HTML data can be dynamically added – though this will be expanded to all formats handled by Ferret shortly.

Figure 5 below illustrates the client-server architecture. The left side of the figure represents the content of the server, while the right side represents the client, an Internet Explorer web browser.

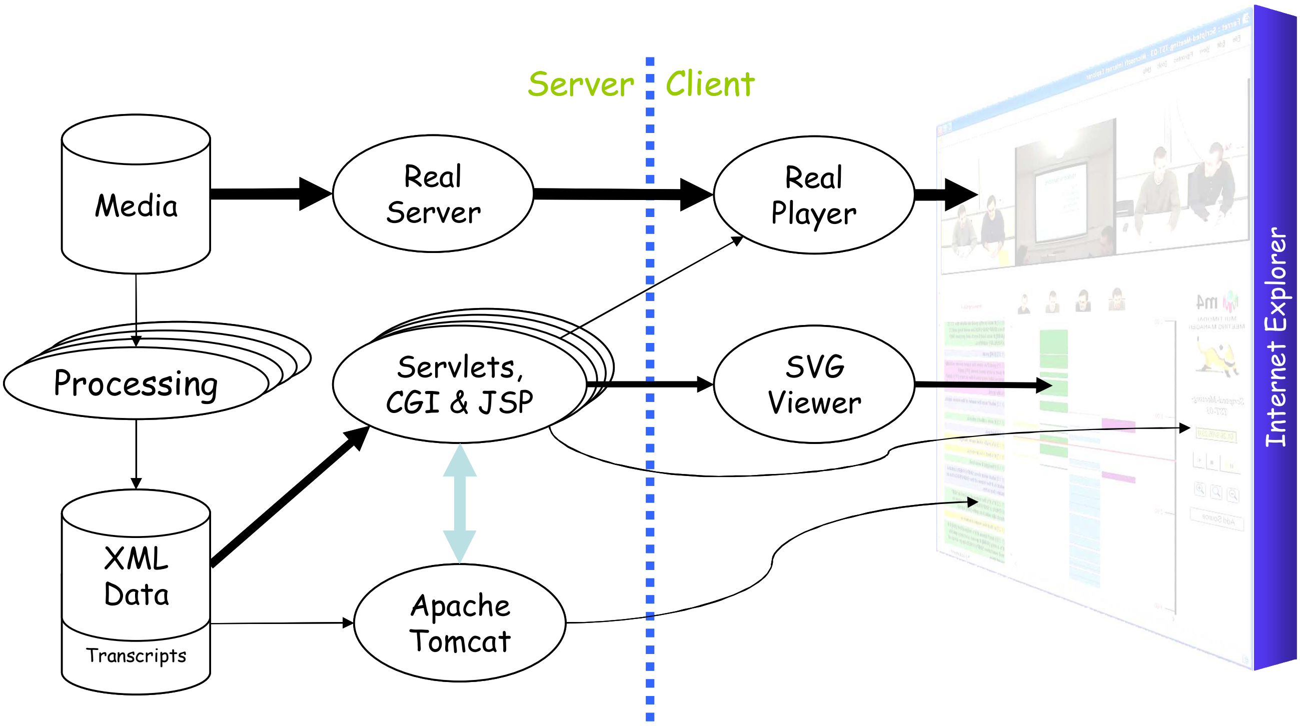


Figure 5 - The Ferret architecture

6.2 Framework

A set of JSP-generated frames is used to produce the overall frame structure of the Ferret browser, dynamically. Each JSP-generated page provides one of the many frames in the display, and is used to create the embedding of plug-ins for video and graphics. In turn, each embedded plug-in is given the URL of data to display.

6.3 Graphical display

A free plug-in, the SVG Viewer (available from www.adobe.com) allows complex graphics to be displayed and dynamically updated. This is used to display the meeting segmentations graphically, and to allow interaction with and dynamic updates of the graphics.

The URL passed by the Ferret framework to the graphics plug-in, invokes a Java servlet. The servlet generates the graphical content from the XML sources, dynamically. Running under the Apache Tomcat servlet container, it reads a list of XML data URLs to display. The servlet fetches each in turn, parses the XML content and generates SVG. Finally, the SVG is streamed back to the graphics plug-in, in the browser, for display.

6.4 Media display

The media resides on the mmm.idiap.ch media server, which contains AVI, RM videos, WAV and RM audio files. It is streamed to the Internet by RealServer and displayed in a browser through the free RealPlayer plug-in (available from www.real.com). RealPlayer facilitates the playback of streamed media, over the Internet, synchronising multiple sources.

The URL passed by the Ferret framework to the media plug-in, invokes a CGI program that dynamically generates SMIL to send back to the plug-in. SMIL is a standard language for presentation of multiple media sources. It allows multiple sources to be played in parallel, or in sequence (or any combination) with time offsets. The video plug-in reads the SMIL and then begins to fetch each media source at the appropriate moment.

6.5 Cookie management

The list of data to display is obtained from cookies kept by the client browser – no session information is kept in Ferret. Every client request to the server is accompanied by the current set of cookies, each giving one source to browse. Cookies are largely managed at the client, by JavaScript functions associated with the various browser forms, including the Add Source dialog, and the source-closing buttons.

6.6 Data processing

Media are used as input for multimedia processing. Various audio, speech, vision analysis algorithms as well as text processing (Awk, XSL) for XML generation are performed. Typically, this is performed ‘off-line’ in batch mode. Ultimately, they all output annotations in XML format. These automatic annotation results from the processing are parsed for display.

6.7 Control

A simple control frame contains player controls, zoom buttons and the “Add Source” dialog access, described below.

The player controls are simple embeddings of RealPlayer plug-in controls.

The zoom buttons simply reload the graphics frame with a new zoom parameter. The list of sources to display is not disturbed, since this is stored as cookies. Therefore, exactly the same set of sources is re-displayed at the new zoom setting. The control panel itself is re-loaded, as new settings for the zoom buttons must be supplied.

The “Add Source” button either creates a new “Add Source” dialog window, or brings any existing one to the front. Entering a URL and pressing the “Add” button creates a new cookie (holding the URL) and forces a re-load of the graphics frame. The graphics are re-generated, now with the new source included.

6.8 Glue Logic

JavaScript functions are used in the browser to link the graphics plug-in, media plug-in, and textual displays, as follows.

The graphical cursor tracks the position of the media plug-in, by polling it every 200ms. Although polling is not recommended generally, it is acceptable in this situation, since it does need to be updated frequently while the player is playing. The polling can be stopped when the player is paused or stopped too. In any case, this proved to be the only reliable mechanism available.

When the cursor is dragged by the mouse, at the moment of release the video-player is asked to move to the corresponding time. From then on, the normal cursor position-polling mechanism takes over. In addition, the transcripts, if any, are asked to scroll to the appropriate position.

If a transcript is clicked, each word or paragraph has an associated time. A JavaScript function causes the media plug-in to play from that time, and asks the graphics frame to scroll to the appropriate offset.

Conversions between times, graphics coordinates and pixel offsets are necessary, along with their inverses – each taking into account the current meeting length and zoom factor.

7 Browser evaluation

Currently there is no standard evaluation procedure for meeting browsers. In previous work, evaluation is either absent or based on informal user feedback e.g. [1]. Where objective data has been collected, user tasks and the questions asked vary widely, are often loosely defined, and final scores are therefore open to considerable interpretation. Most importantly, however, it is not current practice to compare overall meeting browser performance objectively.

In many other fields of research, an objective measure of system performance along with a standard corpus and set of reference tasks has been of enormous benefit in helping researchers compare techniques and make progress. For example, in the field of speech recognition, this has made possible the construction of real time, large vocabulary systems that would not have been feasible ten years ago. The text retrieval conference (TREC) has also used standard corpora, tasks and metrics with great success: average precision doubled from 20% to 40% in the last seven years.

This work aims to develop similar metrics for meeting browsers, and describes a *browser evaluation test* (or BET) for meeting browsers.

There is considerable breadth in what it means to browse a meeting, and in usage scenarios for meeting browsers. For example, the distinction between searching and browsing is not always clear. We consider search for specific events as a part of browsing, but browsing also includes the rapid assimilation of a meeting overview, and the ability to quickly skim through a meeting to find unexpected points of interest. One of the challenges in designing a good browser evaluation test is to create a task that takes into account these multiple dimensions of browsing.

> *We define the task of* ***browsing*** *a meeting recording as an attempt to find a maximum number of* ***observations of interest*** *in a minimum amount of time.*

A key problem in testing browsers, therefore, is identifying these *observations of interest*. The range of possibilities is enormous and depends upon meeting content and individual user interests. The BET method identifies observations of interest based on the impressions of ordinary people. It does not reflect the particular interests of the experimenter or browser designer.

We aim to make the BET:

a) an objective measure of browser effectiveness based on user performance rather than judgment;

b) independent of experimenter perception of the browsing task and meeting structure;

c) produce directly comparable numeric scores, automatically; and

d) replicable, through a publicly accessible web site allowing different researchers to evaluate their browsers and benchmark them.

The following sections first present an overview of the method, and then describes each of its significant features in detail, illustrated by results from a trial evaluation of Ferret using the BET.

7.1 Overview of evaluation method

The BET objectively measures how well a browser satisfies the goal of finding the most observations of interest in the minimum time, using two groups of people: observers and subjects. Observers have no stake in any particular browser, nor any bias about what is interesting in meetings – unlike the experimenters or browser designers themselves. Several observers watch meeting recordings and produce a set of 'observations'. Subsequently, for each browser under test, a fresh set of subjects is presented with questions based on the observations. Ultimately, their answers determine a score for the browser.

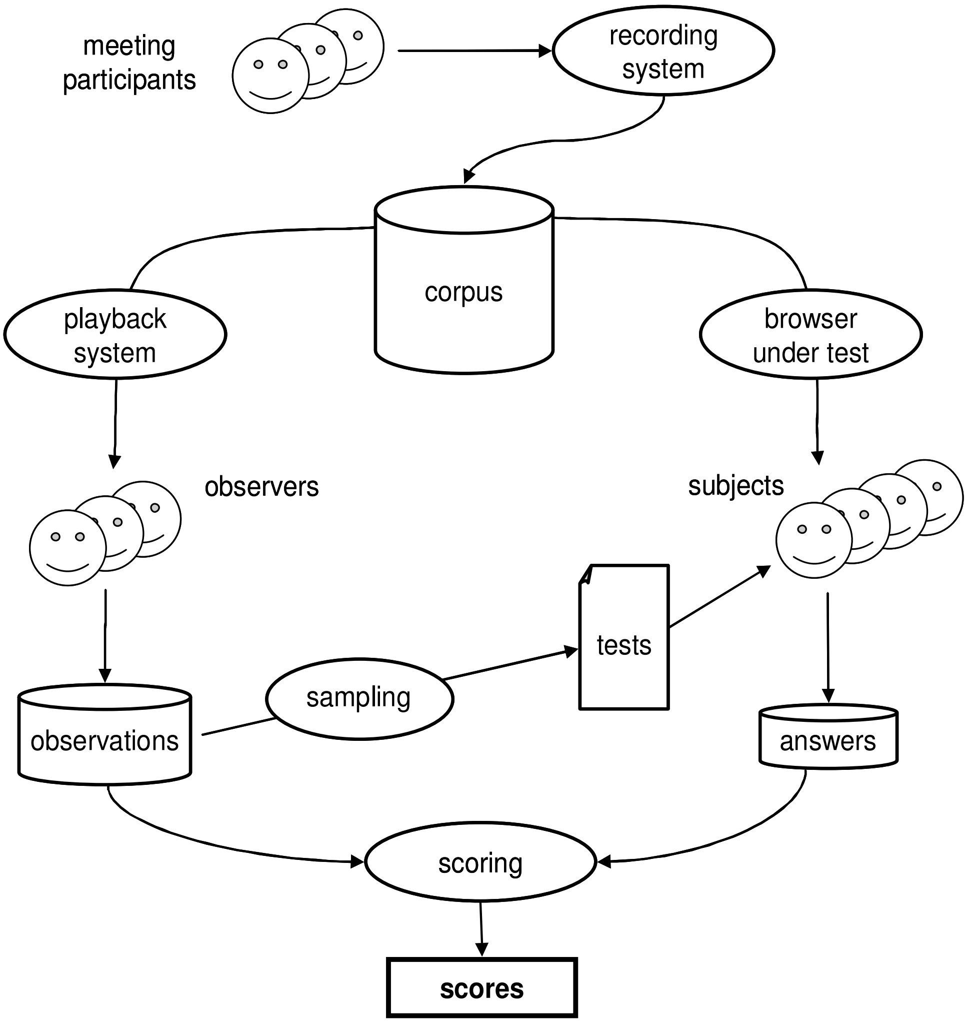


Figure 6. The BET method.

The BET method is summarized in Figure 6 above. The significant features are described below, with further detail in subsequent sections:

- The *corpus* is a significant set of media recordings providing the data to be browsed.
- *Observers* watch selected meetings from the corpus, to produce a store of *observations*.
- Later, during testing, the observations on some meeting are sampled to produce *tests*.

- *Subjects* use the *browser under test* to review the meeting, answering as many test questions as they can in a short time.
- *Answers* produced by the subjects are stored for scoring and analysis.
- *Scoring* compares the subjects' test answers to the original stored observations, to compute a *score* for the browser.

Using the BET requires considerable investment in one-time creation of the corpus and collection of the observations. In order to understand final browser scores, it is also necessary to run benchmark tests for well-known conditions and publish them along with the corpus and observations. This need only be done once, however. Subsequent browser tests take advantage of this one-time work to run tests and produce comparable scores, repeatedly.

Further detail on each of these points is provided in the remainder of the paper. Within each of the following four sections (concerning the corpus, observations, testing and results), the BET method is explained in detail, and then a subsection illustrates its application in a trial evaluation, showing how to construct benchmark tests and scores for a sample meeting browser.

7.2 The Corpus

The corpus is a set of media recordings consisting of the data to be browsed. The BET can be applied to a number of different types of corpus (*e.g.* news videos, home videos, or meeting recordings), but our initial application is meeting recordings.

Design of the corpus has enormous influence on the BET. The corpus determines the observations made, the questions asked, and ultimately the browsing behaviour of the subjects.

BET results obtained with the use of one corpus are therefore not directly comparable to results obtained with another corpus. This implies that a shared corpus must be available to anyone performing comparable BETs, so should not contain sensitive information. It also implies that the relevance of BET scores to real browser applications is dependent on the relevance of the corpus to these applications. For our purposes, the corpus must contain recordings of real meetings. To facilitate the selection of diverse observers and subjects, the content of the corpus should also be comprehensible to a wide audience. Both observers and subjects must be able to follow discussions, reasoning and conflicts within a meeting, although not necessarily in every detail. For example, planning a social event or a common organizational issue is preferable to discussing the mathematics behind a new algorithm.

7.3 Trial evaluation Corpus

The recorded meeting used for the trial evaluation was a 44-minute[2] discussion between four people on how to select and lay out furniture in a university reading room. This recording was made in IDIAP's smart meeting room [10] by A. Lisowska. It is available for viewing (along with all other data discussed in this paper) at the BET web site http://mmm.idiap.ch/bet A large multi-media meeting corpus collection effort (now underway as part of the AMI project http://www.amiproject.org will provide additional meeting recordings for use in future applications of the BET.

7.4 The Observations

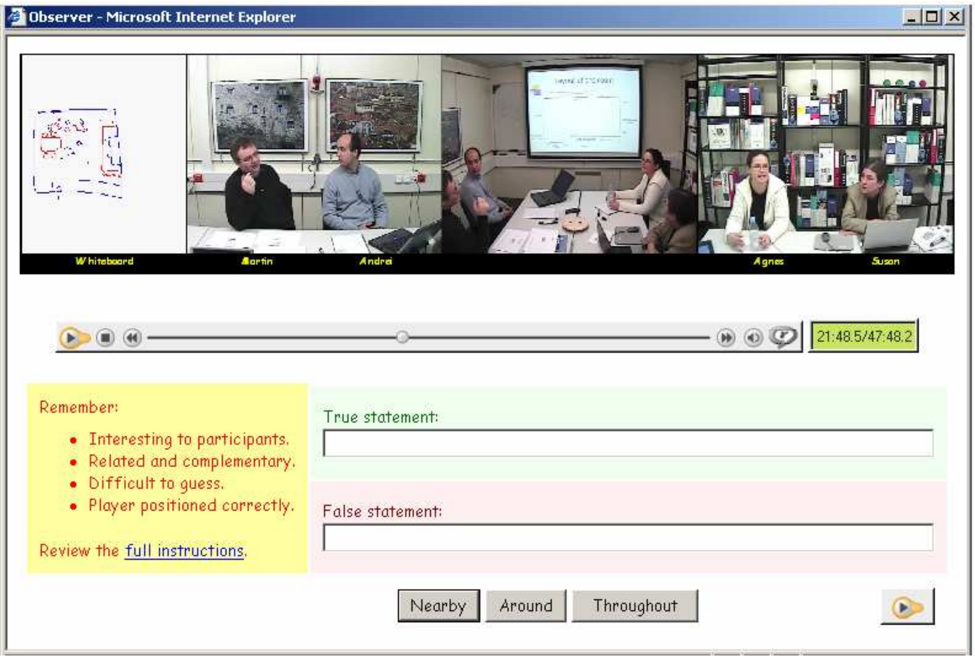
Questions to be used in browser tests are determined by a set of observers, who produce *observations of interest*.

The observers independently (*i.e.* alone) watch selected meetings from the corpus. Observers have available the full recordings from every media source, in parallel, including paper printouts of the slides accompanying the meeting. They may rewind and replay the sources, as they desire. There is no time limit for the observers, but in the trial evaluation, people spent about 4½ times the duration of the meeting to complete their observations.

Instructions are given in a standard manner on a web page made available with the corpus. Each observer is instructed to produce observations that the meeting participants appear to consider interesting. Asking observers to take the perspective of participants is meant to temper undue influence of each observer's own special interests (*e.g.* someone who finds gesticulation more significant than issues discussed). A single observer does not typically make the same observation multiple times, but the most significant features of each meeting are observed multiple times by different people, albeit in slightly different forms. Thus, samples drawn from the set of all observations can include multiple instances of common points of interest, and the statistical distribution of selected observations reflects their relative frequency within the meeting.

This approach avoids the introduction of experimenter bias regarding the relative importance of particular meeting events. Instead of looking for pre-determined general categories of events considered to be significant (*e.g.* agreement, disagreement, action items, *etc.*) we sample from the specific details selected by our independent observers within each particular meeting.

[2] Actually a 44-minute segment from a 47-minute recording.



Remember:

- Interesting to participants.
- Related and complementary.
- Difficult to guess.
- Player positioned correctly.

Review the full instructions.

True statement:

False statement:

Figure 7. Observer web form.

Each observation is stated as a complementary pair of statements, one true and one false, both of which are later presented to subjects during testing. Observers are instructed to produce observations that should not be easy to guess without use of a browser (difficulty is verified later), and the observations should be simply and concisely stated. To encourage brevity, observations are collected via a web form (see Figure 7 above), where the box for the observation text is small.

Observers typically type their true statements first in the upper text area. As soon as they begin typing, the media player is paused so that its position can be recorded along with the observation. To encourage consistency between the two complementary statements, the first statement is automatically copied into the other text field for editing before submission.

Each observation is time-stamped with the media time into the recording, and submitted with an estimate of its locality: *nearby*, *around* or *throughout*. As shown later in the paper, this is used to determine the temporal correspondence between questions and their answers. The observer associated with each observation is recorded, and each observer is given a questionnaire, recording personal and professional details, so that these variables are available to be analyzed for possible influences on the score. (Later, subjects are given a similar questionnaire.)

7.5 Trial evaluation Observations

In the trial evaluation, we collected 294 observations from six observers about a 44-minute meeting, or roughly one observation per meeting-minute per observer. No attempt to verify the observations was made, as this would re-introduce experimenter's judgment – which the BET attempts to exclude.

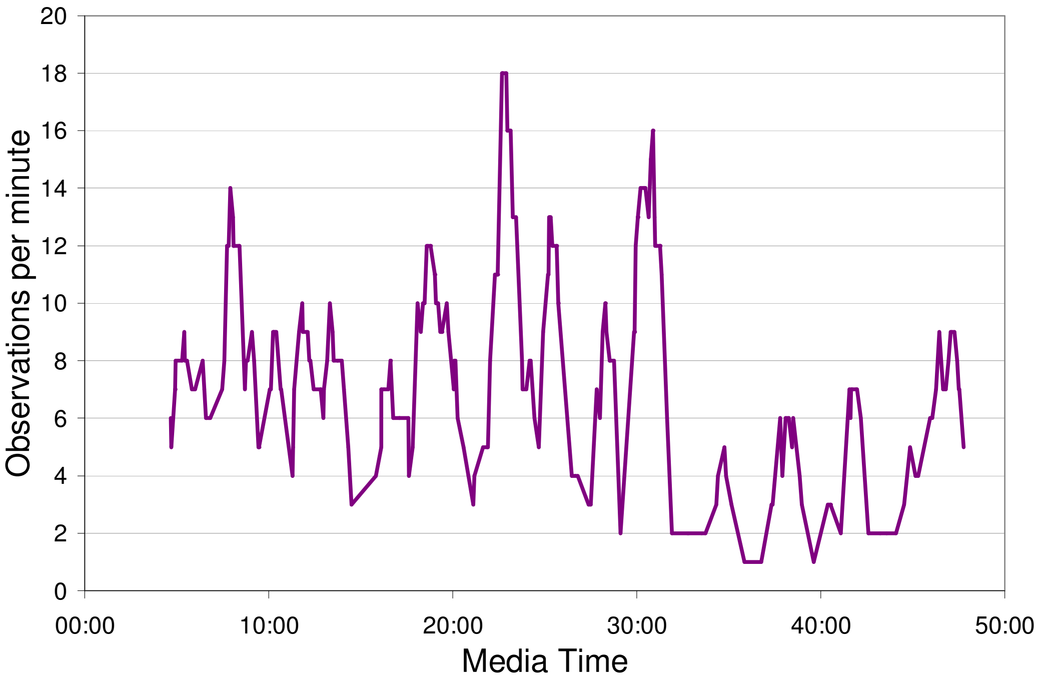


Figure 8. Observation density.

A plot of observation density from the trial evaluation (see Figure 8 above) shows the total number of observations made by all observers within one minute on either side of each observation.

The peaks in this graph identify parts of the meeting that can be interpreted as "hot spots," where the most observations of interest occur in a short period of time. Casual inspection of observations in these peaks shows considerable overlap, *i.e.* most mention the same participant making a point about the same topic. Defining meeting "hot spots" in this way is an alternative method to that used by ICSI [20] but should help browsing performance as defined above, *i.e.* to help people find the maximum observations of interest in the minimum amount of time.

Observations cannot only be used for browser testing, but they can also be used for meeting analysis, and for the development of better browsers in the future. One promising direction, for example, is automatic detection of meeting hot spots using machine-learning techniques such as those proposed in [9]. More hints on useful browser features can be gained by characterizing the observations, especially during peak times. One striking attribute, for example, is that most observations are about individual participants, rather than about overall group actions. Of the top ten most frequently used words in the trial evaluation observations, four were the participants' names, while the remainder were insignificant (*the, of, to, a, is* and *that)*. The name of at least one participant occurred in 81% of our trial evaluation observations (238 out of 294).

These observations support the intuition that spoken words in the audio channel are more important to browser performance than information in the video channels, because most of the observations are about what participants said. For example, the words "want" and "says" appear in the top twenty most frequently used words. This encourages further work on meeting browsers that support navigation through speaker segmentation and speech transcription.

7.6 Browser Testing

Test subjects are neither participants nor observers, and preferably have no direct or vested interest in the content of the corpus. Their task is well defined and effectively determined by the observers, so the precise background and interests of each individual subject is not critical.

Subjects can take several tests, each of which requires them to use the same browser, to examine one of several meetings, one per test. That is, the test is administered "between-subjects" – a necessity, as other researchers may later test other browsers elsewhere. The order in which each meeting is presented is counterbalanced across subjects, to avoid any sequence effect.

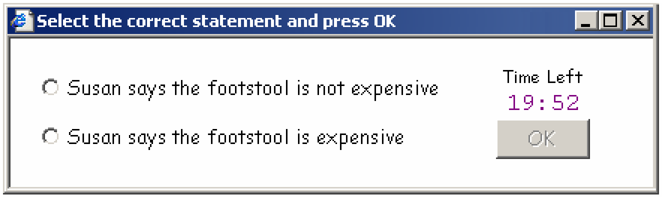
Each test is a set of questions drawn one at a time from the observations. Both the true and false statements of an observation pair are presented together in random order and the subject must use the meeting browser to decide which one is correct. Presenting subjects with both statements, rather than just one, gives them more information about what to look for in the meeting, and highlights the crucial facts necessary to determine the answer.

Questions are presented at the bottom of the screen in a window like that illustrated in Figure 9 below. When one of the statements is selected, the *OK* button is enabled, and when pressed, a new pair of statements is immediately presented for guessing.

Tests have a time limit of half the duration of the meeting under examination. This is partly to simplify scheduling of subjects, but also to prevent a simple playback of the whole meeting from satisfying the questions. Time pressure is required in order to emphasize "the minimum time" stipulation from our definition of browsing. To help remind subjects of their time limit, a continuously running countdown timer is displayed above the *OK* button used to submit answers. Each answer is time-stamped with both the real time of the answer and the media position.

Observations are selected randomly for each test, but no observation pair is used more than once in the same test. In order to avoid a ceiling effect, the number of questions in a test is practically unlimited.

This testing process is entirely automatic with tests administered via the web. This simplifies use of many possibly unknown subjects, but does not imply that any browser must itself be web-based. The media files may be a local copy to maximize playback performance.



Select the correct statement and press OK

○ Susan says the footstool is not expensive

○ Susan says the footstool is expensive

Time Left
19:52

Figure 9. A BET question.

7.7 *Discussion of testing options*

An alternative approach to presenting questions one at a time is to show each subject a large set of questions *all at once*, and ask them to answer as many as possible within the time limit. This approach can be argued to assess the browsing task (rather than just searching) more accurately and may better reflect the scenario where a person is trying to learn as much as possible about a meeting in a short time, rather than an attempt to find one particular fact. However, there are practical problems with this approach (*e.g.* how many questions to present, and finding screen real-estate for hundreds of questions). It also has the disadvantage that it may encourage too much guessing, and that results will vary due to "exam technique" or the ability of subjects to cherry-pick the easiest questions.

However, sequential presentation does indeed test a blend of both searching and browsing effectiveness – not pure searching. During the later part of the test, subjects have already browsed through large parts of the meeting (up to half). Later questions become progressively easier to answer based on the relevance of material viewed while looking for answers to previous questions. This position is at least partially supported by results of the trial evaluation (presented in the next section) which clearly show an increase of speed and accuracy in the later parts of the tests.

7.8 *Benchmark tests*

Published along with each BET corpus and observation set are also two benchmark scores. These are from two one-time tests that are performed using each of the following conditions:

- *Guess condition*: educated guesses with no media present whatsoever;
- *Base condition*: the same basic playback software used by the observers.

The Guess condition reveals whether observations are too easy to guess, and it provides a lower bound below which no browser should sink, no matter how constrained.

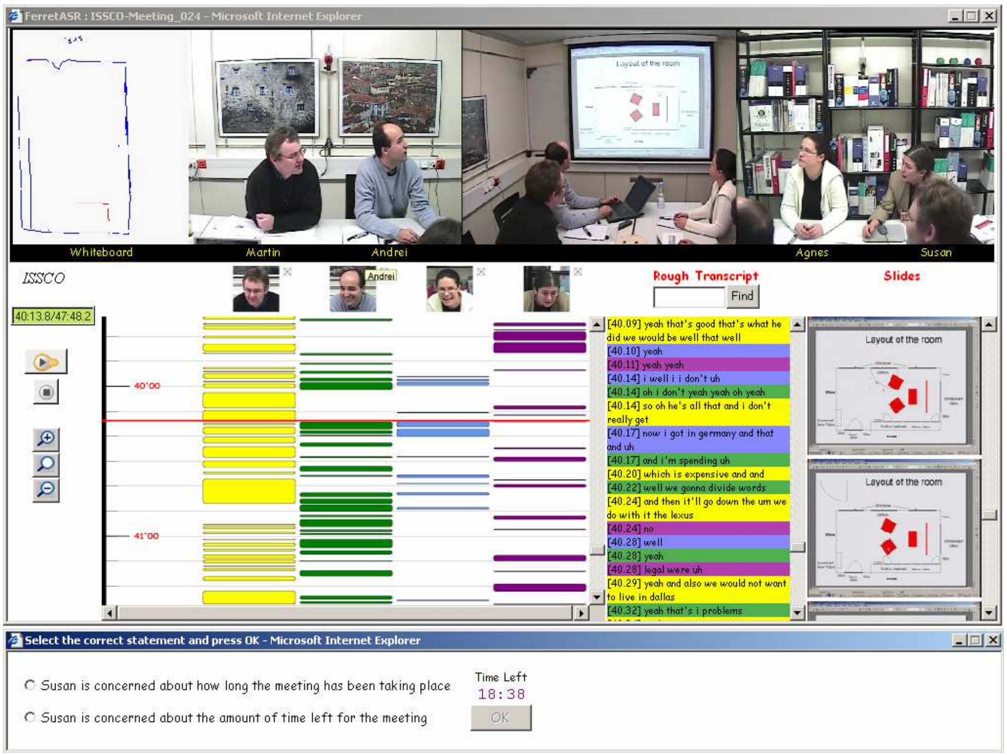
The Base condition provides another useful reference point because we know that all information the observers used was available through this interface, but the observers had unlimited time while the benchmark base test is limited to just half the recording time. A severely restricted browser (*e.g.* video only, without audio) could score lower than the benchmark base, but we would expect most browser designers to consider the Base condition score as a minimum starting point.

7.9 *Trial evaluation tests*

In the trial evaluation, we tested a total of eleven women and thirteen men primarily from academia, whose average age was 35. All subjects were given 22 minutes to answer questions about the 44-minute trial evaluation recording. In the Guess condition, they saw only the question window illustrated in Figure 9, but in the Base condition they also had the media player used by the observers (in Figure 7 above), but without the true and false type-in fields. Eleven subjects were tested in the Base condition, and three subjects were tested in the Guess condition. Guessers worked so fast that they produced more than fifteen times more answers per subject in the Guess condition than in the Base condition, and one subject exhausted the question set. As a result, more subjects were tested in the Base condition so as not to magnify the imbalance in number of answers

7.10 Ferret F_1 configuration

As described above, the experimental Ferret browser can be configured with a range of possible features to assist navigation within a meeting recording. For the trial evaluation, we tested ten subjects using a configuration of Ferret labeled as the F_1 condition, illustrated in Figure 10 below.



[40.09] yeah that's good that's what he did we would be well that well
[40.10] yeah
[40.11] yeah yeah
[40.14] i well i i don't uh
[40.14] oh i don't yeah yeah oh yeah
[40.14] so oh he's all that and i don't really get
[40.17] now i got in germany and that and uh
[40.17] and i'm spending uh
[40.20] which is expensive and and
[40.22] well we gonna divide words
[40.24] and then it'll go down the um we do with it the lexus
[40.24] no
[40.28] well
[40.28] yeah
[40.28] legal were uh
[40.29] yeah and also we would not want to live in dallas
[40.32] yeah that's i problems

Select the correct statement and press OK - Microsoft Internet Explorer

○ Susan is concerned about how long the meeting has been taking place

○ Susan is concerned about the amount of time left for the meeting

Time Left 18:38

OK

Figure 10. The F_1 condition.

The top part of the F_1 screen is the same video and whiteboard player used by observers and the subjects in the Base condition. The bottom part of the screen, however, provides three additional navigation aids: speaker segmentations, a rough transcript generated by automatic speech recognition (ASR), and captured presentation slides, all automatically generated from the meeting recording.

1) The speaker segmentation is presented on a scrollable and zoomable timeline displaying a colored column for each participant whenever that person is detected as speaking. A red horizontal cursor moves along this timeline as the media advances, and users can drag this cursor to control playback position, as well as click on any segment to play it.

2) A very rough ASR transcript generated by Martin Karafiat using the M4 speech recognition system is colored by participant, but has more than a high word error rate. At the top of this column is a text field and "Find" button for searching specific words in the transcript. The user can click on text fragments to move playback to the corresponding point in the meeting.

3) Every slide change that occurred during the meeting is captured and displayed in the right column. Subjects can click on these images to navigate the player to the point in the meeting when that slide was first displayed.

F_1 was tested on people drawn from the same subject pool as the observers and benchmark conditions, primarily at the University of Sheffield. Browser software and media playback was running locally, submitting results to servers at IDIAP. To prevent the possibility of communication lags or browser crashes from invalidating a testing session, the countdown timer resumes from the point of last submission.

8 Results from trial evaluation

Scores from testing the two benchmark conditions and F_1 are presented first as raw scores, followed by three illustrative graphs, and ending with an overall BET score pair for each of the three conditions.

8.1 Raw scores

The score for each subject test is simply the proportion of correct answers obtained. A perfect score for a test would therefore be 100%, while random answers would yield a score of around 50%.

Subject	***Answers***	***Correct***	***Incorrect***	***Score***
A1	255	142	113	55.7%
A2	220	123	97	55.9%
A3	135	81	54	60.0%
Guess Total	*610*	*346*	*264*	*56.7%*

Table 1. Scores for the Guess condition.

Scores for subjects in the Guess condition are summarized in Table 1 above. The three subjects scored an average of 56.7% correct answers. This is consistent with expectations, showing that observations were not too easy to guess correctly. The subjects answered very different numbers of questions – one of the subjects completed all the questions in the database. However, the slowest subject achieved the highest score.

Subject	Answers	Correct	Incorrect	Score
B1	22	14	8	63%
B2	25	17	8	68%
B3	12	7	5	58%
B4	8	8	0	100%
B5	5	2	3	40%
B6	3	1	2	33%
B7	12	8	4	66%
B8	5	4	1	80%
B9	8	3	5	37%
B10	22	12	10	54%
B11	4	4	0	100%
Base Total	*126*	*80*	*46*	*63.5%*

Table 2. Scores for the Base condition.

Scores for subjects in the Base condition are shown in Table 2 above. Of the eleven subjects in this condition, two scored 100%, but one of them with double the questions of the other. Once again, there is greater accuracy at slower speeds. The average score in this condition was 63.5% – somewhat higher than the Guess condition, but with only a fifth of answers. Surprisingly, three subjects scored less than random, despite watching significant portions of the meeting.

Subject	Answers	Correct	Incorrect	Score
C1	20	11	9	55%
C2	6	3	3	50%
C3	18	17	1	94%
C4	21	12	9	57%
C5	18	11	7	61%
C6	11	7	4	63%
C7	6	6	0	100%
C8	14	10	4	71%
C9	12	11	1	91%
C10	7	2	5	28%
F_1 Total	*133*	*90*	*43*	*67.7%*

Table 3. Scores for the F_1 condition.

Scores for subjects in the F_1 condition are shown in Table 3 above. The ten subjects in this condition achieved a score 67.7%. This is larger than the Base condition, and with a slightly larger number of questions answered in the same time (13.3 questions per subject, versus 11.5 for the Base condition).

8.2 Scores over time

There are several times associated with each answer: the real time is recorded, along with the test time remaining for the subject (these are not necessarily directly related, if, for example, a subject needs to switch machines during the test) and the position in the media.

Figure 11 below shows how the average score increases over test time for each condition. The final resting place of the score is that shown in the basic result tables above, with the F_1 condition ahead of the Base condition, ahead of the Guess condition. However, it is interesting to note that both the F_1 and Base conditions were lagging behind the Guess condition for most of the duration of the tests. The gradient of the F_1 score increases significantly with around eight minutes of the test remaining – as subjects become more familiar with either the browser and the meeting itself.

Both the F_1 and Base condition have a final spurt in the last thirty seconds of the test. Intuition and anecdotal evidence suggests that subjects notice their dwindling time remaining, abandon use of the browsers, and simply try to answer as many questions as they can in the final seconds. However, it is interesting to note the high accuracy of these final answers, compared to the earlier answers and the pure Guess condition. This suggests that subjects have learnt much about the meeting content, incidentally, during the test.

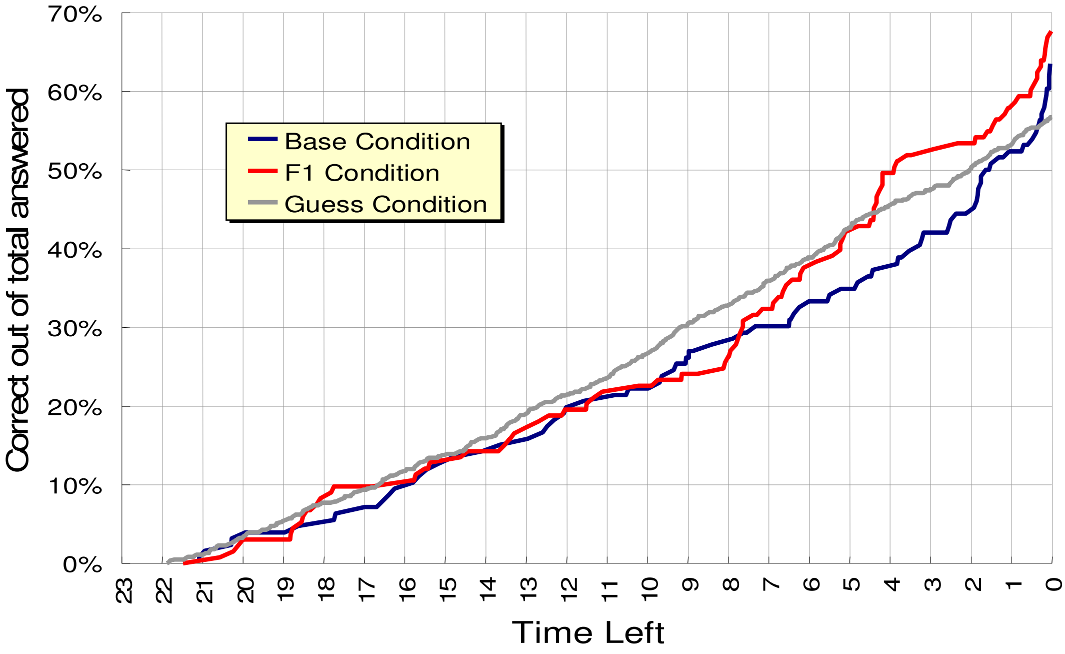


Figure 11. Score increase with time.

8.3 Media time difference

The difference between the media time of the observer's player when the observation was made, and the media time of the subject's player when the answer was submitted is plotted as the *media time difference* in Figure 12 below. On the left side of the graph are answers made before their corresponding questions, while answers made later are shown on the right. Correct answers are counted above the axis, and incorrect answers below.

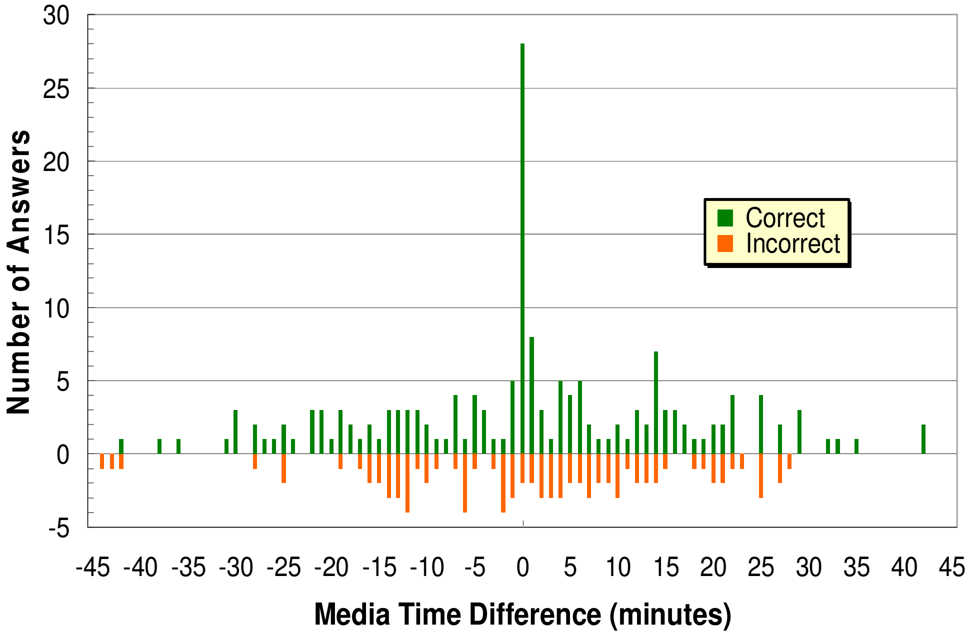


Figure 12. Correct and incorrect answers by media offset.

It can be seen that subjects make many more correct answers (89%) within one minute either side of the original observation[3] – compared to the overall proportion of correct answers (66%). This holds within two percentage points for both the Base and F_1 conditions. The obvious conclusion is that helping users navigate to the correct point in the meeting clearly helps them to answer questions correctly.

[3] Note that a random distribution of questions and answers would naturally yield a simple triangular profile. The small peaks at the extremes are due to answers supplied at the start or end of the recording, concerning the other end of the recording, since the player 'wraps' around at the endpoints.

8.4 Speed versus accuracy

Figure 13 below shows a graph of the number of questions answered by each subject against the proportion answered correctly for the Base and F_1 conditions. Horizontal lines for the Guess condition, the Base condition, and F1 show a progression in accuracy, as expected. The mean values for the two browser conditions are marked, together with one standard deviation on either side on each axis. This shows that F_1 is both faster and more accurate than the Base condition.

It is also evident that both the most accurate and least accurate subjects were amongst the slowest. This suggests that slower subjects were either more diligent, or were presented with more difficult questions. As speed increases, the Base and F_1 subjects tend to become only as accurate as those in the Guess condition. This may be because the browser leads subjects to inappropriate conclusions under pressure, or simply that quickly decided answers degenerate towards guesses.

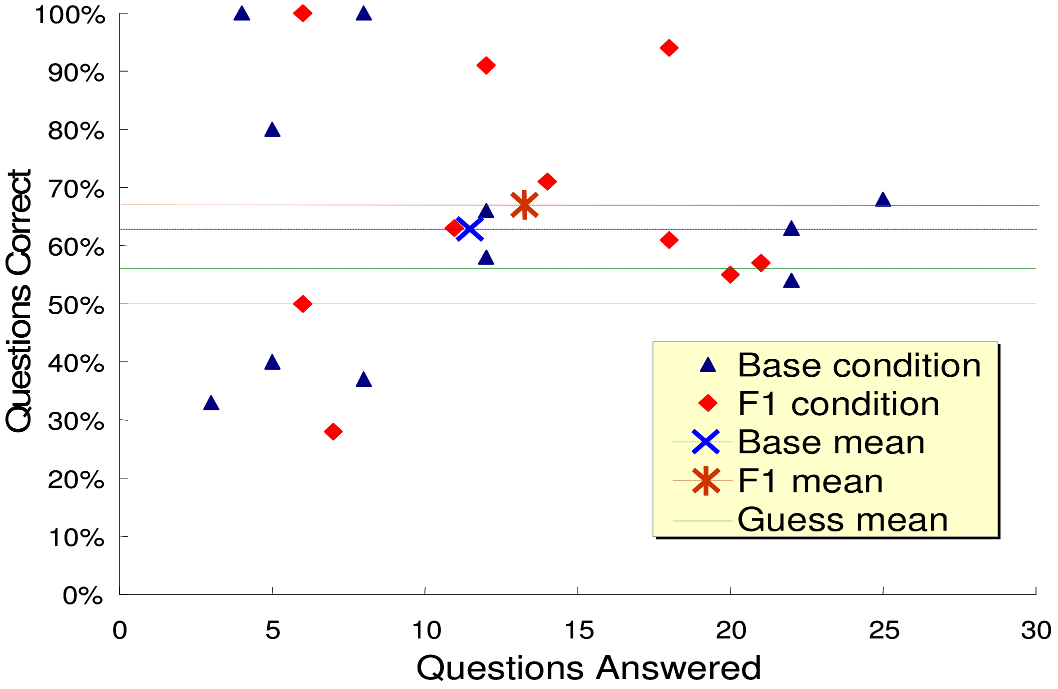


Figure 13. Speed versus accuracy.

The F_1 condition appears to dominate in the mid-range of speeds, where accuracy is highest, with neither a high nor a low speed. The Base condition appears to be either slow or fast, with slower subjects achieving amongst the highest and lowest accuracy, while quicker subjects achieve no better than the Guess condition.

The overall BET scores for each condition are a pair of numbers: one representing the speed of the browser in answers per subject per minute, and the other representing its accuracy, as shown in Table 4 below.

Condition	***Speed***	***Accuracy***
Guess	9.2	56.7%
Base	0.52	63.5%
F1	0.60	67.7%

Table 4. BET scores.

9 Future evaluations

Having carried out this proof of concept demonstration of the BET technique we now plan to extend to larger corpora and to different styles of browser. The AMI project intends to record 100 hours of meetings, which will serve as a corpus for a larger data set, and extend the set of observations available for experiments.

We also plan to use BET to compare different styles of browser, *e.g.* speech only browsers, or browsers with no video stream. Another possibility is to use BET to determine the effect of various quality parameters for the different UI components. For example, we may investigate the effects of ASR quality, or the quality of speaker detection on browsing performance. These comparisons suggest how BET results can be used to improve future system designs. By comparing the BET scores of multiple system designs, we can look at how browsing is affected by various UI components (*e.g.* video, access to slides), as well as quality parameters (*e.g.* ASR, speaker detection). We can then use this information to inform which components are most important for new browsers (*e.g.* video may be unimportant compared with transcribed speech), and which UI components need most improvement (e*.g.* ASR quality). We might also correlate BET scores with logged user behaviors in order to determine whether use of a particular UI feature improved BET browsing scores, again suggesting directions for future designs. Finally, we want to investigate the relationship between BET scores and the subjective evaluations used in many previous studies.

10 Summary & Conclusion

In this report, the important functions of the Ferret browser have been detailed in from a user's point of view, web technology, architecture and data flows.

Ferret itself may be found at http://mmm.idiap.ch/

Further details including detailed software documentation can be found there too.

The browser evaluation test is a method for assessing browser performance on meeting recordings in which the number of *observations of interest* found in the minimum amount of time is used as the metric. Observations of interest are statements about the meeting collected on a meeting corpus by independent observers prior to performing an evaluation. When testing a browser, subjects are presented with questions drawn from the observations, enabling browsers to be scored in terms of both speed and accuracy. A configuration of Ferret including multiple recognition results achieved in M4 showed both higher browsing speed and accuracy than the base condition.

11 References

[1] Ross Cutler and Yong Rui and Anoop Gupta and JJ Cadiz and Ivan Tashev and Li-wei He and Alex Colburn and Zhengyou Zhang and Zicheng Liu and Steve Silverberg, *Distributed meetings: a meeting capture and broadcasting system,* Proceedings of the tenth ACM international conference on Multimedia, 2002, pages 503--512, Juan-les-Pins, France.

[2] Dar-Shyang Lee and Berna Erol and Jamey Graham and Jonathan J. Hull and Norihiko Murata, *Portable meeting recorder,* Proceedings of the tenth ACM international conference on Multimedia, 2002, pages 493--502, Juan-les-Pins, France, ACM Press.

[3] Werner Geyer and Heather Richter and Ludwin Fuchs and Tom Frauenhofer and Shahrokh Daijavad and Steven Poltrock, *A team collaboration space supporting capture and access of virtual meetings,* Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work, 2001, pages 188--196, Boulder, Colorado, USA.

[4] Patrick Chiu and John Boreczky and Andreas Girgensohn and Don Kimber, *LiteMinutes: an Internet-based system for multimedia meeting minutes,* Proceedings of the tenth international conference on World Wide Web, 2001, pages 140--149, Hong Kong.

[5] Chunyuan Liao and Qiong Liu and Don Kimber and Patrick Chiu and Jonathan Foote and Lynn Wilcox, *Shared interactive video for teleconferencing,* Proceedings of the eleventh ACM international conference on Multimedia, 2003, pages 546--554, Berkeley, CA, USA.

[6] Patrick Chiu and Ashutosh Kapuskar and Lynn Wilcox and Sarah Reitmeier, *Meeting Capture in a Media Enriched Conference Room,* Cooperative Buildings, pages 79-88, 1999.

[7] Yong Rui and Anoop Gupta and Jonathan Grudin, *Videography for telepresentations,* Proceedings of the conference on Human factors in computing systems, 2003, Ft. Lauderdale, Florida, USA, ACM Press.

[8] Rogina I. and Schaaf T., *Lecture and presentation tracking in an intelligent meeting room,* in Proceedings of the Fourth IEEE International Conference on Multimodal Interfaces, 2002, pages47—52.

[9] I. McCowan and S. Bengio and D. Gatica-Perez and G. Lathoud and F. Monay and D. Moore and P. Wellner and H. Bourlard, *Modeling Human Interaction in Meetings,* Proceedings of ICCASP, Hong Kong, 2003, April,

[10] D. Moore, *The IDIAP Smart Meeting Room*, 2002, IDIAP-COM 07.

[11] M. Flynn and P. Wellner, *In Search of a Good BET,* 2003, IDIAP-COM, 11.

[12] G. Lathoud and I. McCowan, *Location Based Speaker Segmentation,* Proceedings of the 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-03), 2003, Hong Kong, April.

[13] Daniel Gatica-Perez and Guillaume Lathoud and Iain McCowan and Jean-Marc Odobez, A *Mixed-State I-Particle Filter for Multi-Camera Speaker Tracking,* 2003, IEEE Int. Conf. on Computer Vision Workshop on Multimedia Technologies for E-Learning and Collaboration (ICCV-WOMTEC).

[14] Stephane Marchand-Maillet, *Meeting Record Modeling for Enhanced Browsing,* Technical Report, University of Geneva, March 2003.

[15] W3C *SVG*, Scalable Vector Graphics, http://www.w3.org/Graphics/SVG/Overview.html

[16] W3C *SMIL*, Streaming Media Integration Language, http://www.w3.org/Graphics/SVG/Overview.html

[17] M4 (IST-2001-34485), *D3.3: Final Report on multimodal information access and integration*, 2005.

[18] H. Lee et al, *The Físchlár digital video recording, analysis, and browsing system*, in Proc. Content-based Multimedia Information Access (RIAO'2000), Paris, France, 12-14 April 2000.

[19] CMU Informedia, http://www.informedia.cs.cmu.edu/

[20] A. Janin, J. Ang, S. Bhagat, R. Dhillon, J. Edwards, J. Marcias-Guarasa, N. Morgan, B. Peskin, E. Shriberg, A. Stolcke, C. Wooters, and B. Wrede, The ICSI meeting project: Resources and research, in ICASSP 2004 Meeting Recognition Workshop.

[21] P. Wellner, M. Flynn, M. Guillemot. Browsing Recorded Meetings With Ferret, In Bengio S. & Bourlard H., eds. (2004), MLMI'04: Proceedings of the Workshop on Machine Learning for Multimodal Interaction, LNCS, Springer-Verlag, Berlin.

[22] P. Wellner, M. Flynn, S. Tucker, S. Whittaker. *A Meeting Browser Evaluation Test*, in ACM Conference on Computer-Human Interaction 2005 (CHI 2005).