



FP6- 506811

**AMI
AUGMENTED MULTI-PARTY INTERACTION**

<http://www.amiproject.org/>

Integrated Project
Information Society Technologies

**D6.3 PRELIMINARY DEMONSTRATOR OF
BROWSER COMPONENTS AND WIRELESS
PRESENTATION SYSTEM**

Due date: 31/06/2005

Submission date: 31/06/2005

Project start date: 1/1/2004

Duration: 36 months

Lead Contractor: PHI

Revision: 1.1

Project co-funded by the European Commission in the 6th Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	<input checked="" type="checkbox"/>
PP	Restricted to other programme participants (including the Commission Services)	<input type="checkbox"/>
RE	Restricted to a group specified by the consortium (including the Commission Services)	<input type="checkbox"/>
CO	Confidential, only for members of the consortium (including the Commission Services)	<input type="checkbox"/>



D6.3 PRELIMINARY DEMONSTRATOR OF BROWSER COMPONENTS AND WIRELESS PRESENTATION SYSTEM

Bram van der Wal (PHI)
Jamie Walch (VNC)
Pierre Wellner (SPI)
Mike Flynn (IDIAP)
Jan Cernocky (BRNO)
Pavel Zemcík (BRNO)
Stanislav Sumec (BRNO)
Simon Tucker (SHEF)
Thomas Kleinbauer (DFKI)
Job Swiers (TWENTE)

Abstract: This document describes the work done for WP6 in the first 18 months. This includes preliminary versions of the AMI browser components and the integration of the Wireless Presentation System. Section 2 describes the Philips Wireless Presentation System integration with the meeting browser using RealVNC technology. Section 3 describes the work done by University of Sheffield regarding audio plug-in. Section 4 describes work done so far by Spiderphone regarding a Spiderphone JFerret plug-in. Section 5-7 describes the work done by Brno University regarding the Key-word/audio-event spotting engine JFerret plug-in, the Virtual environments and the Automatic video editing plug-in. Section 8 describes the work done by University of Twente regarding the Virtual Meeting Room plug-in. Section 8 describes DFKIs summarization plug-in.

Content

Introduction	2
1. Browser components integration.....	3
2. Philips - Wireless Presentation System (WPS)	4
WPS integration with meeting browser using RealVNC technology	9
3. University of Sheffield - Audio plug-in	13
4. Spiderphone - Spiderphone JFerret plug-in	14
5. Brno University of Technology - Keyword spotting JFerret plug-in.....	15
6. Brno University of Technology - Collaborative Data Sharing	17
7. Brno University of Technology - Automatic video editing plug-in	18
8. University of Twente - Virtual Meeting Room	19
9. DFKI - MASCOT (Meeting browser Abstractive Summarization Component).....	20
Conclusion	25
References	26

List of figures

Figure 1 JFerret plug-in architecture	3
Figure 2 Abstract AMI demonstrator architecture	3
Figure 3 WPS connected to VGA monitor	4
Figure 4 Token-based control over PPT renderer	5
Figure 5 Wireless Presentation System functional overview	6
Figure 6 Philips Wireless Presentation System PPT renderer	7
Figure 8 Application showing UPnP radar & two UPnP devices with advertised services	8
Figure 9 Different application views	9
Figure 10 WPS integration using RealVNC	9
Figure 11 RealVNC technology inside Adderlink IP	9
Figure 12 PPT slide plug-in of JFerret	12
Figure 13 Spiderphone JFerret plug-in	14
Figure 14 Search window for keyword spotting.	15
Figure 15 Results of keyword spotting in an archive of meetings.	16
Figure 16 Browser window with keyword spotting panel and ASR panel.	16
Figure 18 Architectural overview of ABSURD	21
Figure 19 A user interface design study for MASCOT	23

Introduction

In Deliverable 6.1 'Preliminary demonstrator architecture and API specification' all members of WP6 presented their plans on what and how they would like to implement their research and technologies into the AMI meeting browser. To show the progress that has been made in the mean time this Deliverable 6.3 has been written.

Philips integrated its Wireless Presentation System with the Slide module of the JFerret browser using RealVNC technology. Sheffield developed an audio plug-in, which will soon be integrated as a JFerret plug-in. Brno University developed a key-word spotting JFerret plug-in, a virtual environments plug-in and an automatic video editing plug-in. University of Twente worked on a Virtual Meeting Room application. Spiderphone developed a JFerret plug-in for its web-based Spiderphone application. DFKI developed a summarization JFerret plug-in.

Section 2-8 describe these applications in more detail.

1. Browser components integration

In WP6 we extended the Meeting Browser application from IDIAP, called JFerret, with several plug-ins. Below you see an overview of these plug-ins. Most plug-ins are directly fed into the JFerret application as Java plug-ins. At the time of writing this document not all plug-ins have been completed for integration, so their for we describe the most complete integration modules in the following sections.

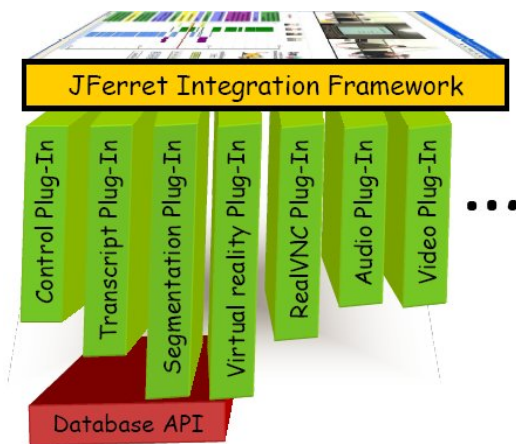


Figure 1 JFerret plug-in architecture

For the integration between Wireless Presentation System (WPS) and JFerret RealVNC technology is used by Philips. This will be described in detail in section 2.

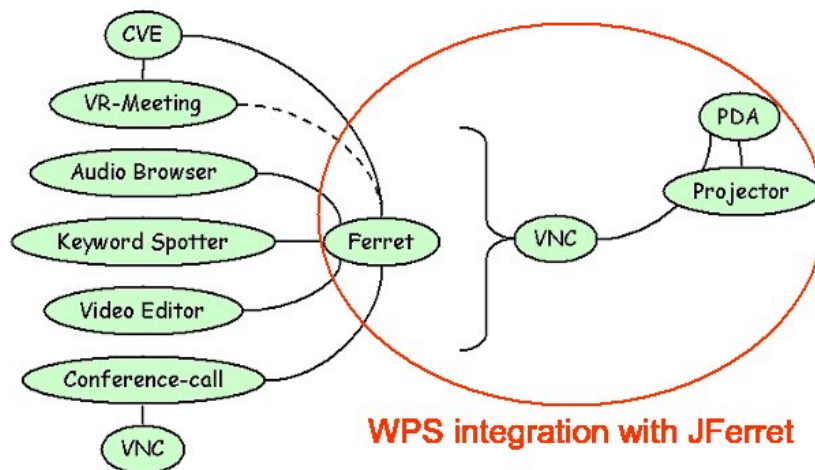


Figure 2 Abstract AMI demonstrator architecture

2. Philips - Wireless Presentation System (WPS)

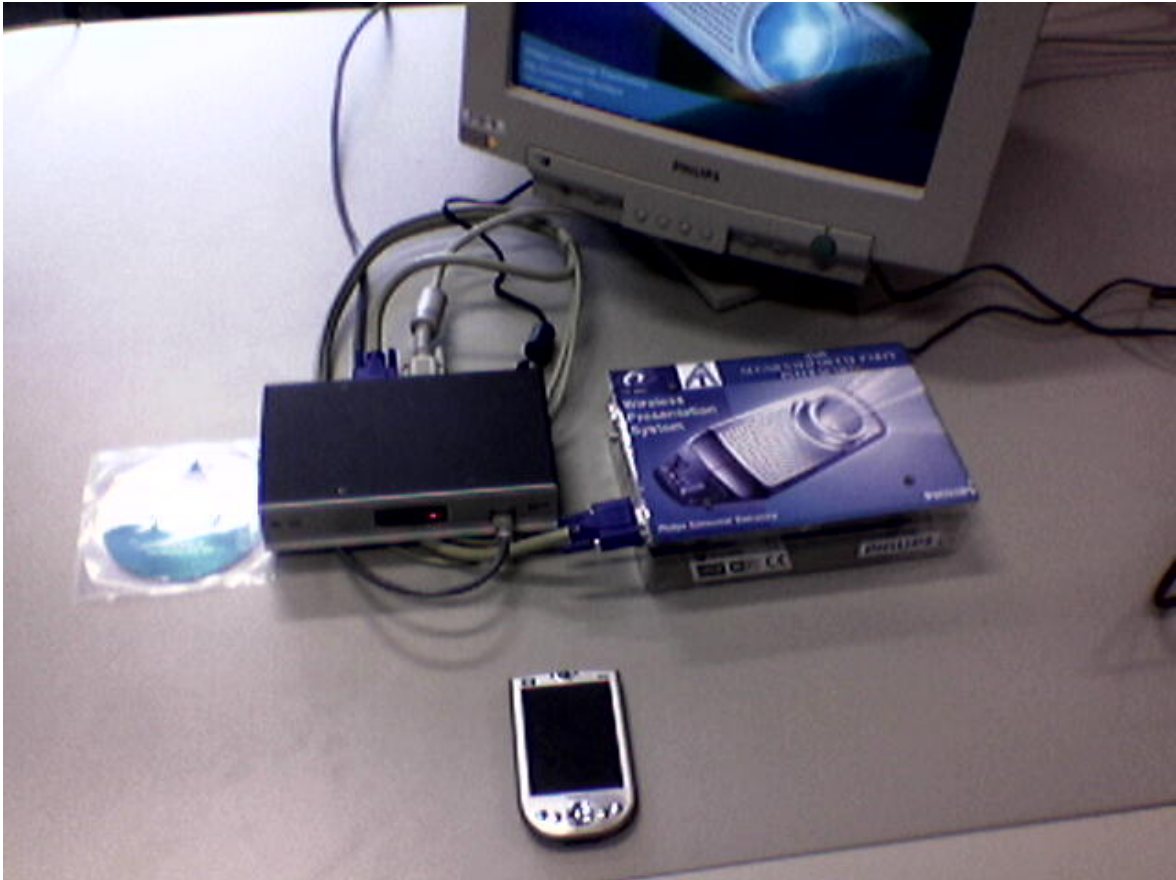


Figure 3 WPS connected to VGA monitor

During the MLMI05 event in Edinburgh in July 2005 Philips showed its WPS and the corresponding AMI integration parts to the public, AMI partners and European Commission reviewers.

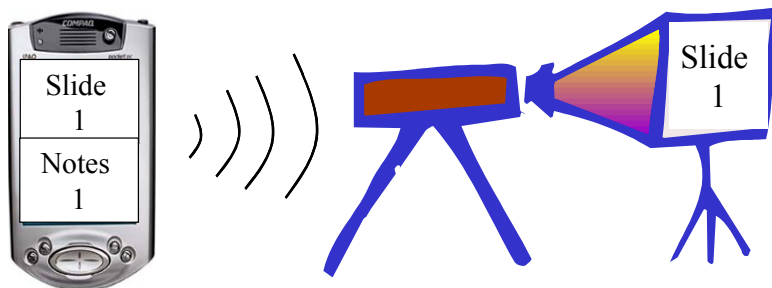
The WPS consists of a wireless enabled (WiFi - IEEE 802.11b) PDA running Pocket PC 2003 from Microsoft and serves as the portable presentation helper of a person that gives a PPT presentation. The WPS also exists of an embedded PPT rendering module, which can wirelessly receive PPT files from the PDA and display them on a monitor or projector by the VGA connector.

With the envisaged application, a presenter could spontaneously connect his mobile device (e.g. PDA) to the projector in a meeting room and then give a presentation from his mobile device. The selection of the wireless projector is token-based. If a PDA is using the Projector to display a PowerPoint presentation, other PDA users will get a message that the Projector is in use by another PDA.



Figure 4 Token-based control over PPT renderer

The mobile device is used to *control* the *presentation* and to *support* the speaker during the presentation by providing a private display for reviewing notes, showing the next slide in turn, etc.



UPnP technology in summary

UPnP technology is all about making home networking simple and affordable for users so the connected home experience becomes a mainstream experience for users and great opportunity for the industry. UPnP architecture offers pervasive peer-to-peer network connectivity of PCs of all form factors, intelligent appliances, and wireless devices. UPnP architecture leverages TCP/IP and the Web to enable seamless proximity networking in addition to control and data transfer among networked devices in the home, office, and everywhere in between. UPnP technology can be supported on essentially any operating system and works with essentially any type of physical networking media - wired or wireless - providing maximum user and developer choice and great economics.

Messages are formatted using the Simple Service Discovery Protocol (SSDP), General Event Notification Architecture (GENA), and Simple Object Access Protocol (SOAP). The above messages are delivered via HTTP, either a multicast or unicast variety running over UDP, or the standard HTTP running over TCP. Ultimately, all messages above are delivered over IP.

References:

<http://www.upnp.org/>

Wireless Presentation System functional description

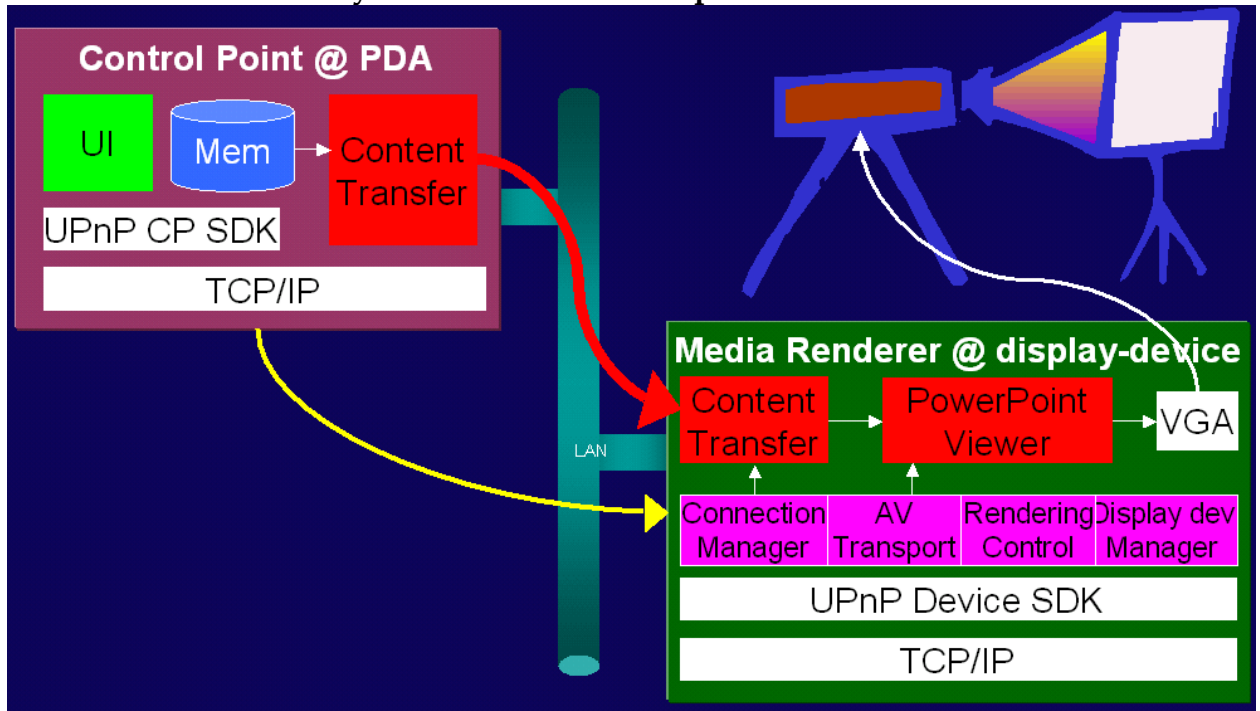


Figure 5 Wireless Presentation System functional overview

UPnP technology is used to automatically detect, connect and control the wireless presenter application from a PDA or laptop. UPnP technology enhances the possibility that a user is able to walk in a meeting room and search for available UPnP services (e.g. Wireless Presenter). With the presentation application (Java-based) running on the PDA or laptop, available projectors are automatically being advertised so you select the preferred projector to present on.

Java-based PDA application

The handheld application is fully based on java.awt and java 1.1.8 (Personal Java). We have installed Jbed CDC Java Virtual Machine on the PDA so we are able to run our own developed presentation application. This application is capable to detect UPnP (Universal Plug and Play) enabled devices on the wireless LAN network. An UPnP enabled device could be a wireless PDA, a PC/laptop or for the purpose of AMI an UPnP enable wireless presentation system (simulating a IP network enabled projector with a built-in PowerPoint renderer).

Embedded UPnP PowerPoint renderer

The PPT renderer box is a fiberglass box with inside a mini PC board running the Windows CE 4.1 platform. This mini PC board is equipped with a WLAN 802.11b WiFi card.



Figure 6 Philips Wireless Presentation System PPT renderer

The scheme below shows the main software modules of the WPS PPT renderer.

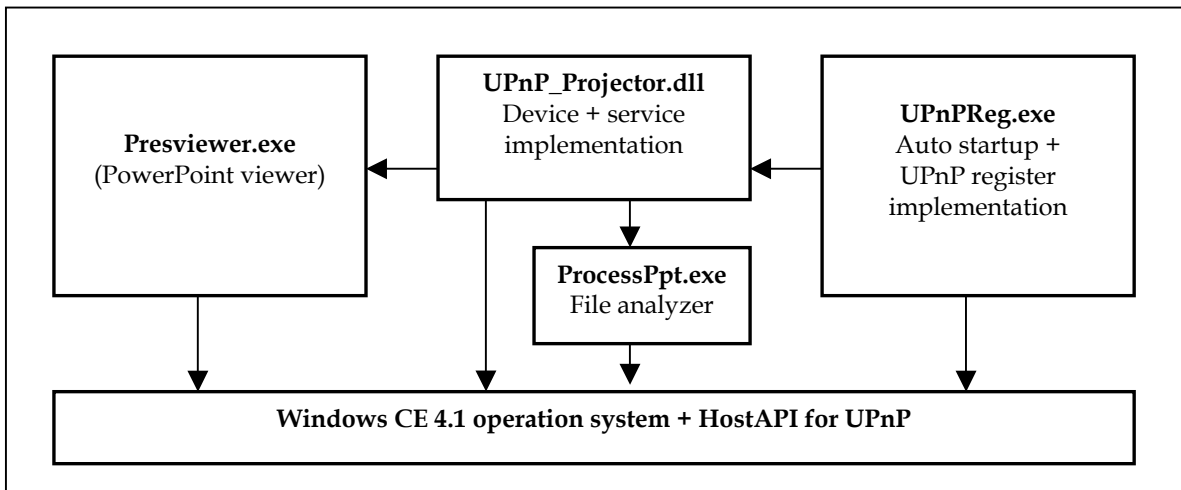


Figure 7 Sub modules of embedded UPnP renderer device

UPnP_Projector.dll is the kernel of the UPnP device. In Microsoft environment UPnP services have to be implemented as COM objects.

UPnPReg.exe is used to register the device dll in Microsoft environment. The exe is called on startup.

Presviewer.exe, **Imageviewer.exe** are usable viewer modules.

ProcessPpt.exe is an internal module to get some information out of the PowerPoint presentation.

Additionally the UPnP HostAPI and a standard Web Server is used.

WPS User operation steps

The following steps will show how the user can operate the Java application on the PDA to support him when giving a presentation:

1. The user can search for and select the wireless network in his environment. When connected to a wireless network the user can select an available PowerPoint renderer (like Philips UPnP wireless Projector) and start an application that is advertised by that UPnP device.

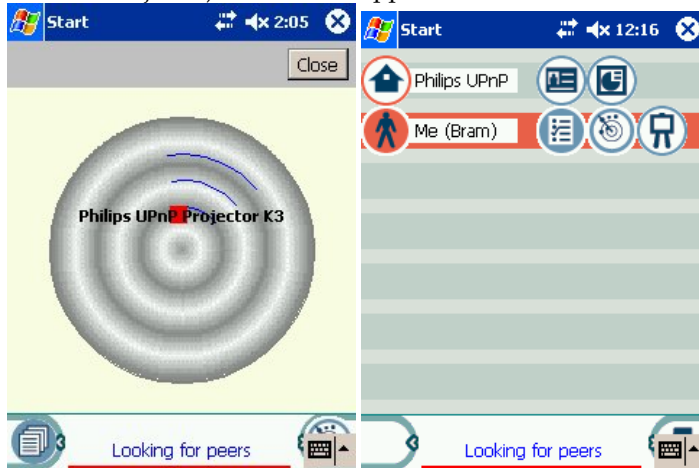
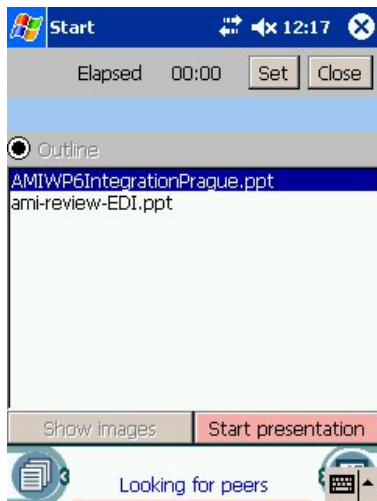


Figure 8 Application showing UPnP radar & two UPnP devices with advertised services

2. When the user has started the PPT application the user can select the PowerPoint file on his PDA that needs to be transferred to the PowerPoint Renderer.



3. The original PPT file on the PDA is wirelessly transferred to the PowerPoint renderer device.
4. The PowerPoint device renders all PPT slides in a memory cache
5. The user can control (next, previous, browse) the PPT slides, which are stored on the rendering device from the handheld using the touch screen of the PDA. Also PPT slide comments can be read on PDA to assist the presenting person with additional information.



Figure 9 Different application views

WPS integration with meeting browser using RealVNC technology

Philips worked together with RealVNC and IDIAP to integrate the WPS with the meeting browser of IDIAP. An AdderLink IP box (using embedded VNC) is inserted between the PowerPoint renderer and the projector, see Figure 10. This would allow remote users (or those who prefer to watch the presentation on their laptop) to see the slideshow in real time over an IP network.

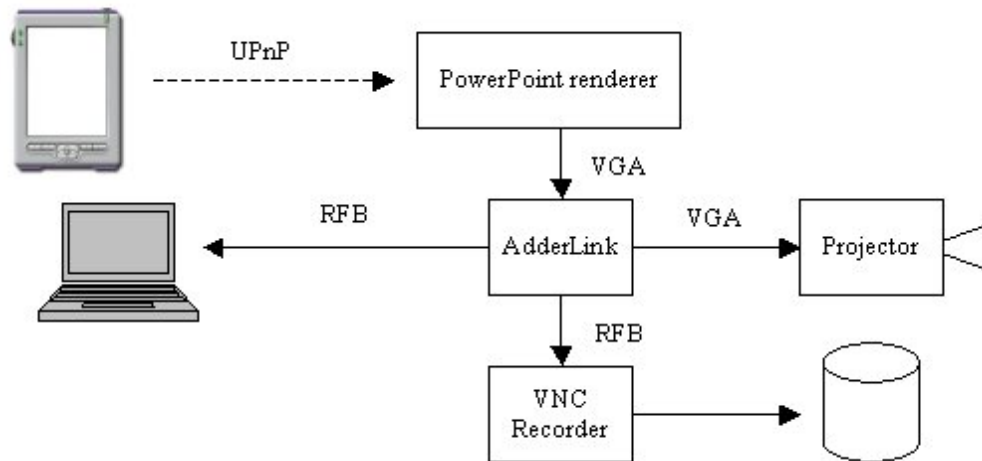


Figure 10 WPS integration using RealVNC

The Adderlink digitizes all VGA input passing through it to an RFB stream, which it makes available like any other VNC Server. If the RealVNC recording proxy is attached to the Adderlink, it will record the slide changes. However, it will not 'know' that a slide change has occurred, so you would not be able to index the recording by slide number. However slide animations will be recorded. It would probably be possible to post-process the recording to add meta-data.



Figure 11 RealVNC technology inside Adderlink IP

For the integration with IDIAPs meeting browser we implemented 2 types of RealVNC PC platform applications, which will be described below:

- VNC recording suite
- VNC snapshot

VNC recording Suite

The VNC recording suite consists of 2 modules.

One module for recording the RFB stream (vncrecorder.exe) and one module to play the recorded RFB stream (vncplayer.exe). This recording application records a complete video sequence (RFB) of all VGA data.

vncrecorder.exe

This is recording application. In its most basic use, you run it with the following command:

```
vncrecorder kfr=50 <host>[:<display>]
```

This will connect to the VNC server running on <host>, which must be configured for 'No authentication' and any encryption mode other than 'Always On' (if you are worried about security, you can configure the server not to accept inputs, or to accept connections only from the machine on which you are running the recorder). The display number is usually 0 and can be omitted in this case. The KFR=50 sets the key frame rate to 50%; the larger this value is the larger the recording will be on disk, but the better you will be able to seek within the file during playback.

The VNC recorder will output three files: an XML control file, an FBL file containing the graphical data, and a KFL file containing key frame data. By default, these are named according to the hostname and the date and time. You can change the XML output file on the command-line as follows:

```
vncrecorder kfr=50 <host> output=<filename>
```

default.xml

This is a configuration file that is read by vncrecorder and specifies things like the name of the computer, location and user for a recording. Most of these are just copied verbatim to the XML output file, but the kfl and fbl options are the names of the KFL and FBL files described above.

These are usually specified as 'default', which causes a unique filename to be generated.

You can have more than one configuration file and specify which one to use on the command-line to vncrecorder using the Configuration parameter. If none is specified the file default.xml from the working directory is used.

vncplayer.exe

This is the playback application. You run it with the following command:

```
vncplayer play=<file>
```

Where <file> is the the XML file generated by vncrecorder. vncplayer is VNC server that listens on port 5995 by default; you connect to it using a standard VNC viewer and connect to desktop number 95. It will start playing immediately and you can control the playback by clicking on the desktop to bring up a control panel. You can also control it by sending appropriate messages as described in the technical documentation.

VNC snapshot application

To overcome the problem of the huge recorded video data to be stored on disk and the off-line slide indexing that would be needed to process the slides inside the meeting browser we developed the VNC snapshot application. This application saves the current VGA output into a GIF picture when a certain threshold percentage (default 5%) of pixels have changed.

If you are connecting to an AdderLink IP, then there may be noise in the signal;

You can tweak how sensitive VNC snapshot is to this via the tolerance parameter. If this value is too low, you may find paradoxically that nothing is saved at all. This is because each noisy pixel causes an update that resets the MergeUpdateDelay timer; if you're unable to find a suitable tolerance to solve the problem, try reducing MergeUpdateDelay instead.

Below you see the description of the parameters of the snapshot application. It starts like this:

vncsnapshot.exe

```
vncsnapshot <host>[:<port>]
```

It will connect to the specified host and save GIF image files to the current directory. If you want to tweak the process, you can specify the following options:

-OutputDir=<dir>

Where to save the XML and GIF files. Default = (current directory).

-GifColors=<n>

Number of GIF palette entries. Default = 512.

-GifDither

Enable dithering of the generated GIFs. Default = off

-Threshold=<percent>

Set the percentage of pixels that must change in order to save a snapshot. Pixels that do not trigger a change will be saved and counted again next time an update occurs. Default = 5

-Tolerance=<amount>

The amount by which a pixel must change to count towards the threshold. This is the sum of the differences in the R, G and B components, with white to black being a change of 3*65535. Default=0

-MergeUpdateDelay=<milliseconds>

The length of time to wait after a significant update (defined as being one that has at least one pixel changing by more than the tolerance and which is not completely contained within previous, unprocessed, updates) before processing it. If you find that you frequently get snapshots being taken in the middle of a window being redrawn, increase this value. Default = 500

The VNC snapshot application generates a XML file with a list of GIF files and the correct order in time. This XML file can be loaded into IDIAPs JFerret Slide plug-in (see Figure 12, right panels).

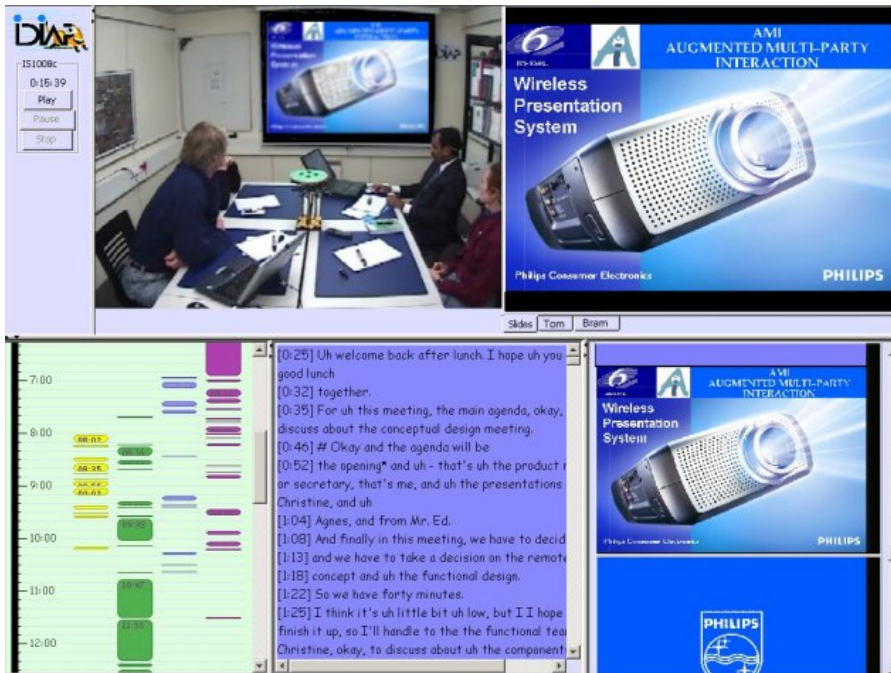


Figure 12 PPT slide plug-in of JFerret (right panels)

Below you see an example of the content of such a XML file:

slides.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<Cards name="media" attribute="time" xmlns="ch.idiap.jferret.plugin">
<Slide src="slide00.gif" target="media" attribute="move" value="0000000"/>
<Slide src="slide01.gif" target="media" attribute="move" value="0048770"/>
<Slide src="slide02.gif" target="media" attribute="move" value="0095620"/>
<Slide src="slide03.gif" target="media" attribute="move" value="0119840"/>
<Slide src="slide04.gif" target="media" attribute="move" value="0124750"/>
<Slide src="slide05.gif" target="media" attribute="move" value="0135060"/>
<Slide src="slide06.gif" target="media" attribute="move" value="0142470"/>
<Slide src="slide07.gif" target="media" attribute="move" value="0218650"/>
<Slide src="slide08.gif" target="media" attribute="move" value="0260990"/>
<Slide src="slide09.gif" target="media" attribute="move" value="0356990"/>
<Slide src="slide10.gif" target="media" attribute="move" value="0422170"/>
</Cards>
```

3. University of Sheffield - Audio plug-in

University of Sheffield (UoS) developed an audio plugin, which alters the playback rate of a speech recording whilst keeping the pitch of the speaker constant. It uses an implementation of the SOLA algorithm to achieve the alteration. The plug-in works in near real time and can produce a range of levels of compression. UoS also has produced Matlab code, which examines the transcript of the recording and produces word and utterance level excision on the speech recording according to measures of word and utterance importance (we are essentially retaining the most important word and utterance). The code produces a new recording at a specified level of compression and also constructs a mapping between the two timescales (compressed time and normal time). This allows anyone using the compressed recording to know where in the uncompressed recording they are.

UoS has this working in real time as a Media Plug-in for the Java Media Framework currently. We also have an experimental JFerret plugin, which allows real time switching between the pre-compressed recordings, and the normal recordings as described above. The JFerret plug-in replaces the original audio player in JFerret (hence the experimentalness) and ensures that other JFerret components (although it does not work with the video recordings) are informed of the underlying time of the compressed recording.

UoS also has a small demonstrator application, which uses the precompressed speech recordings and allows the user to switch between the different compression techniques in real time.

In the future the real time plug-in could be included in JFerret and we could also consider constructing real time components, which make use of the excision-based techniques (which our experiments have shown to be superior to the speech rate alteration approach).

4. Spiderphone – Spiderphone JFerret plug-in

Spiderphone has processed its recordings of audio, PowerPoint, and Active Talker Events, into XML files compatible with JFerret plug-ins. The attached screenshot demonstrates how a Spiderphone recording can be browsed in the JFerret AMI browser by speaker segmentation and by slides. At the same time a little window at the top shows who is the currently active speaker.

In this case, there is no video, and the speaker segmentations are produced by measuring which phone line is the loudest. It is a good example of how the JFerret browser components can be used to browse meeting recordings captured from a source other than the AMI meeting rooms.

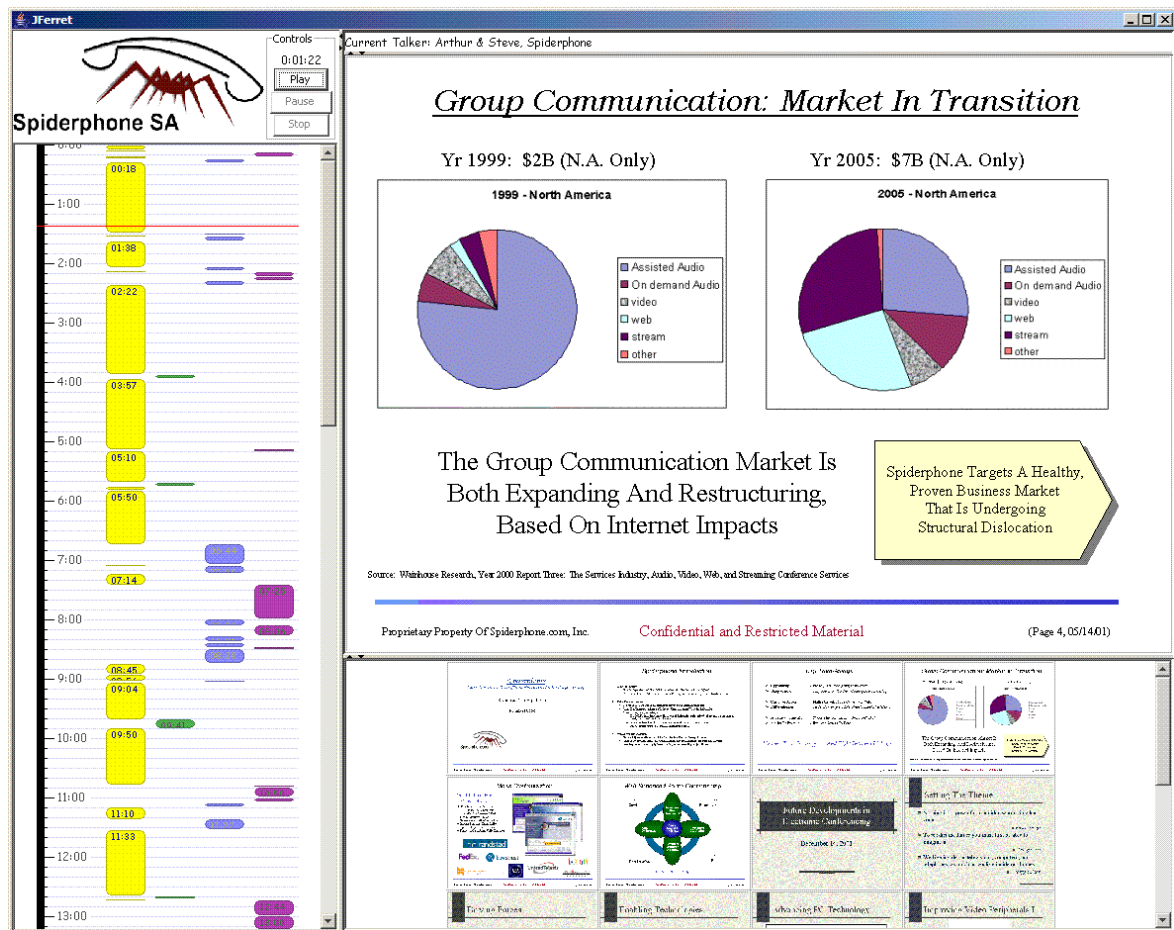


Figure 13 Spiderphone JFerret plug-in

5. Brno University of Technology – Keyword spotting JFerret plug-in

One of tasks of BUT in AMI is to provide the project with keyword spotting (KWS) in meeting environment. We are working on several approaches to KWS including searching large vocabulary continuous speech recognition (LVCSR) lattices, acoustic search and a hybrid “phonetic” search [Szoke2005].

We have developed an indexation and search engine allowing for fast keyword search in LVCSR recognition lattices [Fapso2005]. This engine allows for searching in an archive of multiple meetings, and is able to handle also more complex queries than a simple keyword search (Figure 14).

The search engine was integrated with JFerret:

- * a search window similar to Google can be used to search in an archive of meetings (Figure 14).
- * a list of hypothesis (sorted according to word confidences) is returned and the user can click the occurrences (Figure 15).
- * when clicked, a standard JFerret window is shown, with sorted hypothesis of that particular meeting in one sub-window (Figure 16, lower-right panel).

The plug-in is fully integrated with the rest of JFerret, so that when a hypothesis is clicked, all the other information streams (timeline, audio, video, slides, ASR transcript) get the information about the timing of the hypothesis and synchronize themselves.

We have also completed JFerret by a simple word search – a “grep” - in automatic speech recognition (ASR) output (Figure 16, lower-center panel). The user can use both modes. Both ASR output and recognition lattices were generated by the AMI-LVCSR system [Hain2005-II].

The search engine and JFerret integration are currently available in stand-alone mode (a windows executable is called from the Java code, which in turn processes its results). A networked version is being tested, where a client (JFerret) connects to a search-server holding all the indexes and recognition lattices and returning only the results to the browser.

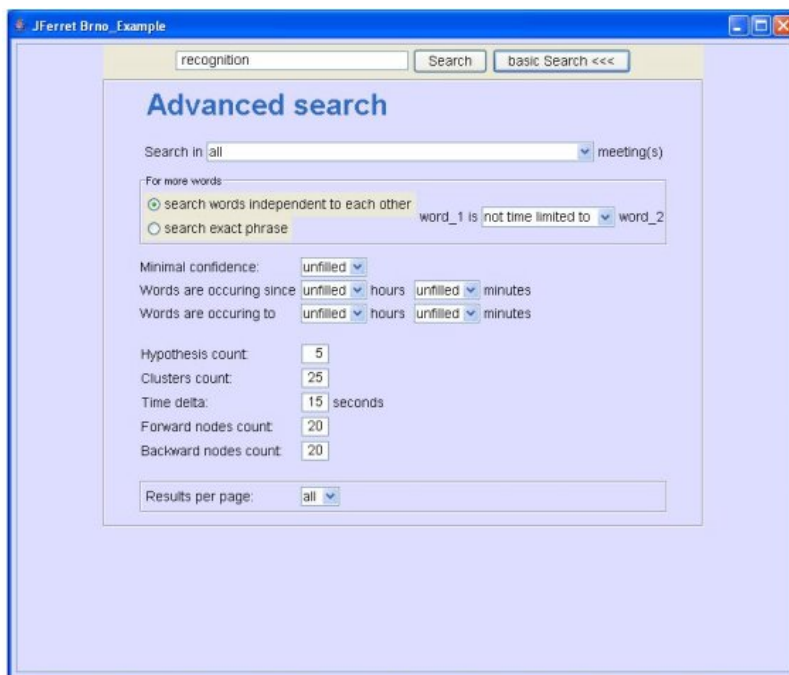


Figure 14 Search window for keyword spotting.

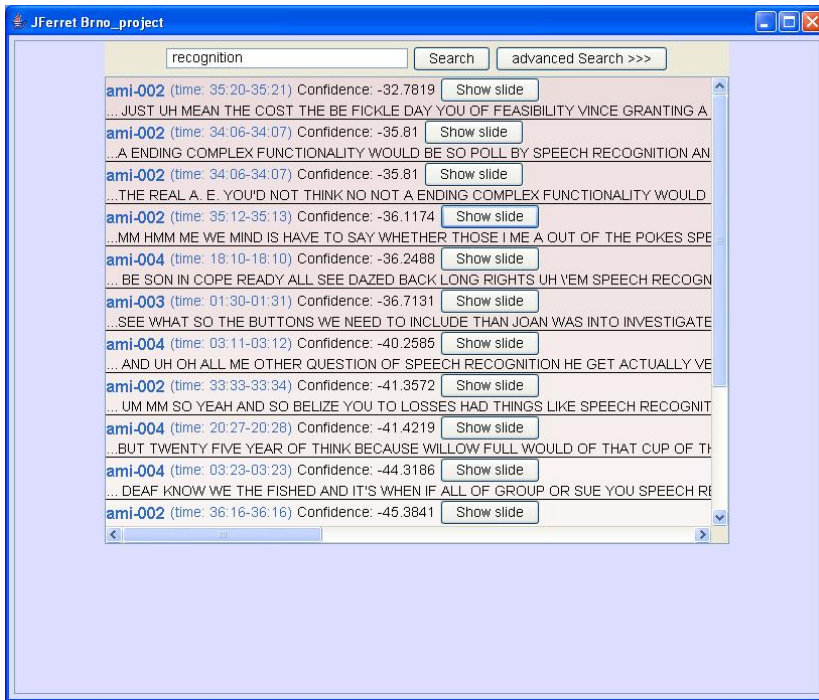


Figure 15 Results of keyword spotting in an archive of meetings. The color corresponds to confidence of hypothesis.

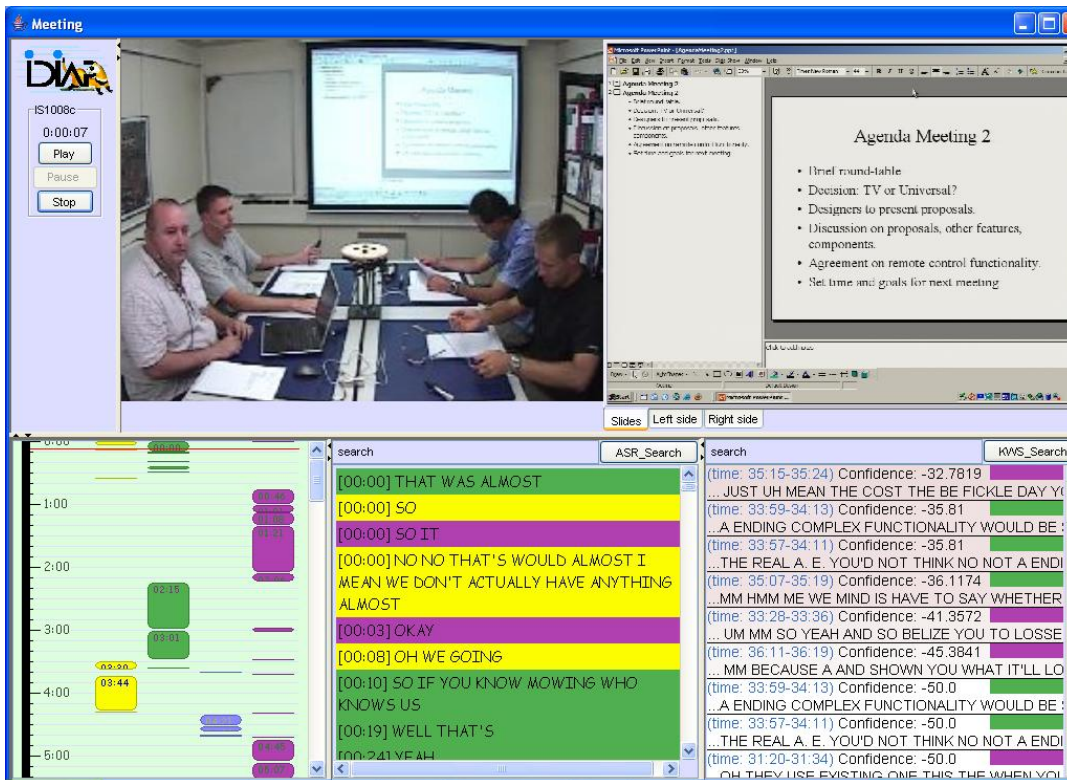


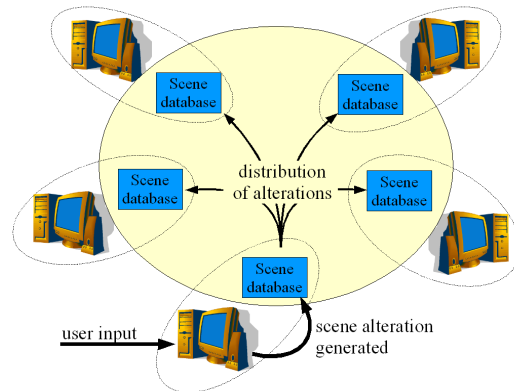
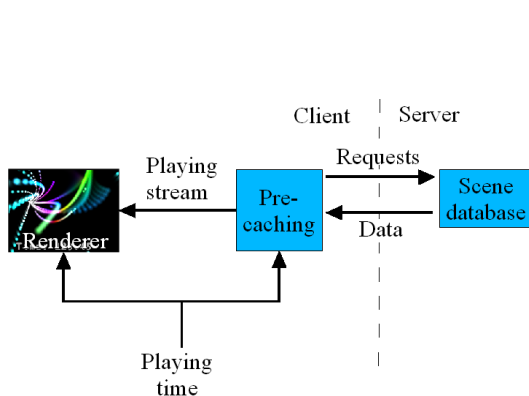
Figure 16 Browser window with keyword spotting panel (lower-right) and ASR panel (lower-center).

6. Brno University of Technology - Collaborative Data Sharing

Collaborative Virtual Environments (CVE)

CVE system provides the meeting environment users a chance to share data, e.g. a 3D model, modify it, and discuss the data with higher quality of interaction comparing to "just video" especially when used in remote meetings [Pe2004].

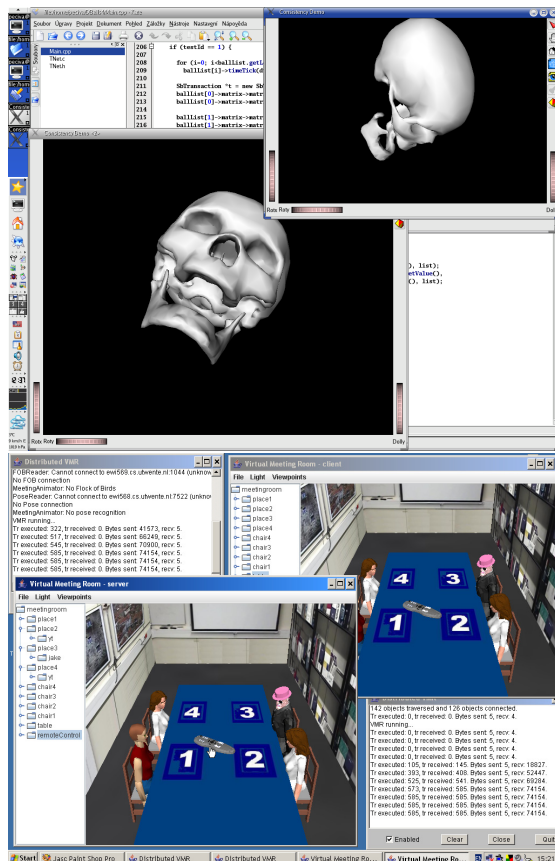
The scene alterations are recorded on the meeting server, enabling later replaying in the meeting browser.



Offline mode: Replaying scene alterations

Online mode: All users see alterations of others

The online usage of the CVE module allows for viewing and low-latency modifications of e.g. 3D scene so that all the users can immediately see effects of the changes and possibly they can also share the viewpoint. The offline browsing mode allows not only for recording of the changes but also for replaying the exact view of the shared data the meeting participants were using [Be2005].



Screenshot of an application for medical data sharing: Long network latency may cause scene to be temporarily inconsistent. Such inconsistencies are noticeable just on large distances (Europe - Japan) or terrible network conditions. The inconsistencies are immediately reconciled.

Distributed Virtual Meeting Room example: Virtual meeting data acquired from head-trackers and body-pose recognizers can be used for creation of virtual meeting room - the technology is investigated by University of Twente. Utilizing our data sharing technology, the virtual meeting room can be used simultaneously by many people from distant areas. All remote participants are represented by their avatars and they are participating in virtual meeting in the same way as local participants.

7. Brno University of Technology - Automatic video editing plug-in

The automatic video editing plug-in (AVEP) provides an automatic video editing of multiple video sources into one edited video output that is transmitted to remote meeting participants through a streaming server. Such edited video output relieves the users from watching all video streams from remote meeting room and additionally saves the network bandwidth. Also, one projector can be used for projecting of the video from the remote meeting room instead of potentially more video projectors or multiple small views on a single projector. Significant part of this topic (e.g. multimodal feature extraction, evaluation of camera selection, application of film editing methodology, video streams processing, video summarization, etc.) is included in WPs 4 and 5 [Po2004][Su2004]. The impact of AVEP on WP6 is mostly in the plugin for RMA whose experimental version is currently available. The edited video stream is transmitted through RTP protocol using JMF (Java™ Media Framework). Such stream can be received on the client side e.g. using Java applet (AVEP) included in a web page - it is easily integrated into the RMA [Su2005].

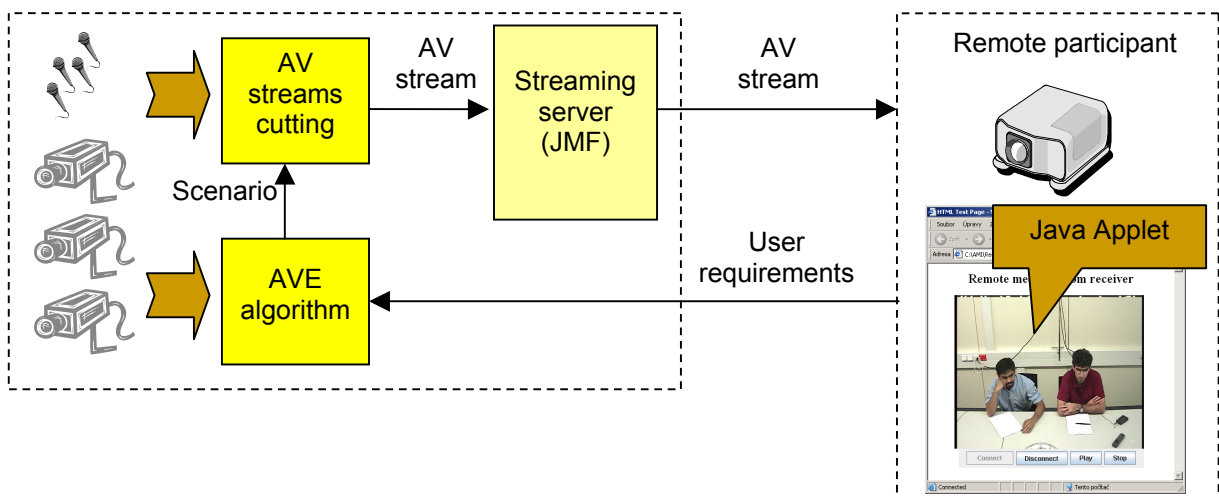


Figure 17 Automatic video editing architecture

8. University of Twente - Virtual Meeting Room

A prototype of a Virtual Meeting Room (VMR) has been developed. Currently, the focus of attention is *replay* and *analysis*. A JFerret plug-in is under development that can show a meeting from *arbitrary* viewpoints, even “through the eyes” of one of the meeting participants. The plug-in can replay recorded and annotated AMI data, where the timeline is under control of the JFerret browser.

A second way of using the VMR is in assisting remote meetings. A first experiment has been done, in cooperation with Brno University, where a remote participant is presented, together with other participants, in a joint meeting room. The remote participants have a virtual position at the table, and can watch the meeting from that virtual position or, if they prefer, can watch the meeting from a more global point of view. The same hold true for the other participants: they will see the remote person at his or her virtual position, making the movements and gestures of the real person. The technology is based upon simple consumer web cams, together with image recognition technology that extracts key features, like body position and gestures.



Remote participant, making some gesture.

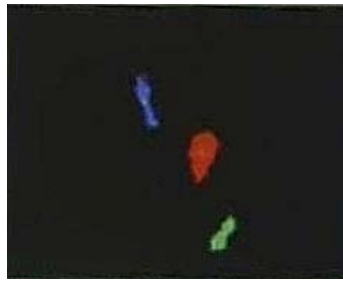


Image and gesture recognition



Remote participant as seen through the eyes of participant number 2.



All participants, some of which are not representing real AMI researchers, seen from a global viewing point

The JFerret plug-in currently exists as an early prototype. Due to the technology that is necessary for showing 3D images within a Java based JFerret browser, the available JFerret Media Player could not be used;

Some work has been done in order to create an extension that is able to show Java3D images as a panel inside the JFerret browser, together with other plug-ins.

9. DFKI - MASCOT (Meeting browser Abstractive Summarization Component)

Introduction

To extend the JFerret browser for user access to a summary of the currently loaded meeting, DFKI is currently developing MASCOT, the abstractive summarization component. The summary is automatically generated from the meeting recordings, thereby utilizing a number of other components developed within the project.

In order to generate such a summary, the component fully analyzes the propositional content of the meeting. The analysis results, together with other annotations such as dialog acts, are used to create an internal representation of the meeting's discourse. This is what distinguishes the abstractive summarization approach from extractive summarization: while the latter tries to apply statistical methods to find important utterances in the speech transcripts of a meeting, the abstractive approach attempts to build up an "understanding" of the discourse. The internal, symbolic representation of this understanding can then be used to regenerate in written form what is considered relevant by the summarizer. As the NLG generation process can draw from a complete and quasi-semantic representation of content, the output can be expected to outperform extractive summarization with regard to text coherence, fluency and readability.

It is perspicuous that the rich modeling of concepts and roles required for this approach cannot be carried out for arbitrary meeting contents. Therefore, AMI chose to limit the abstractive summarizer to the hub scenario "design of a remote control".

The following sections describe DFKI's plans on developing this component, explains architectural ideas and describes an interface design study for the integration into the browser. In addition, the connection to other AMI partners' work is highlighted.

Overview

MASCOT consists of two major sub-components, each of which can itself be divided in multiple interacting modules: the summarizing engine (ABSURD) and a JFerret compatible user interface (U.I.). The latter is being realized by means of JFerret's highly modular plug-in architecture and allows for a fine-grained configuration of summarization parameters (see details below). These parameters are passed to ABSURD, an architectural overview of which is depicted in figure 1.

This representation is passed to the "Information Reduction and Reorganization" component, which finds irrelevant parts of the discourse, given the user's understanding of relevance. It also performs a restructuring of the chart representation, leading to unique and atomic information units that hold all information to be included in the final summary.

Generation

In principle, this internal representation of the summary contents opens up the possibility to generate a number of different summary formats: it installs the prerequisite to pay respect to such factors as user preferences, the user's situational context, target platform constraints (e.g. in the case of the remote meeting assistant), etc. In MASCOT, the primary way to display the abstractive summary is in purely textual form.

To put this into practice, a distinct component -- the "Document Planner" -- is being implemented. For MASCOT's user interface component, a first version of the Document Planner will structurally conform its output to WP3's summarization annotation manual for abstractive hand-annotations [AMISUM] that was used for the recordings of the hub corpus. This manual was handed out to the meeting participants for them to fill out a short template with four slots. For a more detailed description, please refer to the section about MASCOT's draft user interface.

For the actual text in such a summary, the document planner first prepares the structure of the document together with instructions on how the informational content of the summary's internal representation has to be realized. These instructions together with the pre-structured summary skeleton are then passed to ABSURD's "Realizer" component where the information units are finally transformed into surface forms, i.e. words and sentences. In this process, the Realizer not only asserts that the resulting text contains all required information, it is also responsible for a coherent text flow and high readability.

As mentioned before, each of the sub-components consists of multiple interweaved software modules; for reasons of brevity, however, the above description abstracted from technical details.

User Interface Design Study

For the integration of ABSURD output into JFerret, our group has conducted a initial design study for seamless integration as a JFerret plug-in. Here, two major design issues have to be addressed: the limited amount of screen space for display of a possibly lengthy amount of text, and the realization of configuration facilities for the user to adapt the summarization to her personal needs and preferences. Figure 19 shows the result of this design study in the lower right corner of the screenshot.

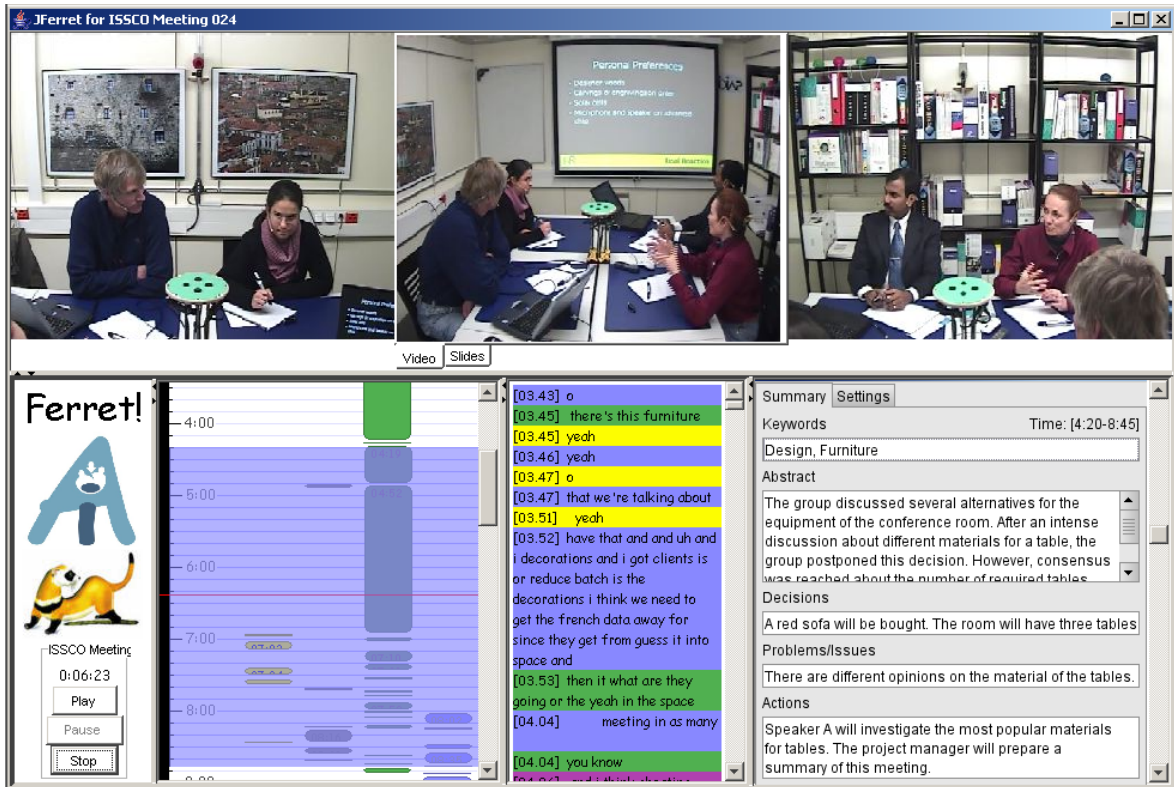


Figure 19 A user interface design study for MASCOT

In accordance to [AMISUM], the U.I. draft consists of four slots taken from WP3' abstractive summarization annotation manual: one abstract paragraph summarizing the recent meeting, one slot for all decisions made in the meeting, one slot to name problems that occurred, and one slot to list individual actions that each of the participants was assigned to take before the next meeting.

These slots are arranged in one common panel, so that the user can observe all of them at a glance. While the three lower slots present their contents in fragmented notes, similar to information extraction systems, the main summary can be found in the "Abstract" slot as a coherent text.

But of course, MASCOT's U.I. is not limmited to this particular implementation of WP3's annotation manual. As figure 2 shows, it is reasonable to add more slots, like e.g. a "keyword" slot, which depicts important terms from the meeting's discussions. For a first, concrete implementation, a filtered re-use of UEDIN's and TNO's topic detection components is aspired.

Currently, the AMI group at DFKI is planning to cooperate with their in-house interaction design group to evaluate the user interface and ensure that it is intuitively usable.

Connection to other AMI work

The description of the architectural overview already mentioned some of the components developed by other partners of the AMI project which ABSURD builds upon. Most of them -- if not already available - are expected to be up and usable for ABSURD in the next months. To sum them up again, the following components will contribute information to ABSURD:

- Automatic Speech Recognizer ASR (WP4)
- Named Entities Detection (UShef)

- Dialog Act Labeling (ICSI, UEDIN, DFKI)
- Speaker & Addressee Detection (WP4, UT)
- Hotspot Detection (ICSI)
- Emotion Recognition (WP4)
- Meeting Acts (ICSI, IDIAP)
- Topic Segmentation (UEDIN)

(The acronyms in parenthesis name the work packages and/or partner institutes working on the referring components).

Conclusion

This deliverable described the WP6 integration work done for the first 18 months of the AMI project. In the coming months the development of more JFeret plug-ins is expected as indicated by some of the WP6 partners. At the end of 2005 WP6 partners will aim to show an AMI meeting browser demonstrator, which will consist of integrated technologies by all WP6 partners. In the mean time all AMI partners are free to test the already available WP6 plug-ins and Wireless Presentation System for evaluation.

References

[Alexandersson2003] Alexandersson, J. "Hybrid Discourse Modelling and Summarization for a Speech-to-Speech Translation System", Phd thesis, Universität des Saarlandes, Germany, December 2003, http://www.dfki.de/~janal/public_archive/thesis.pdf

[AMI D5.1] "Report on Initial Work in Segmentation, Structuring, Indexing and Summarization", Deliverable D5.1 of the AMI project, <http://www.amiproject.org>

[AMISUM] "Summarization Annotation Manual", AMI internal document, <http://www.amiproject.org/private/WP03/annotgroups/summ/Files/Summarization-Manual.ps>

[Be2005] Beran Vítězslav, Pečiva Jan: AUGMENTED MULTI-USER COOPERATIVE ENVIRONMENT, In: Proceedings of the 11th conference STUDENT EEICT 2005, Brno, CZ, FEKT VUT, 2005, p. 528-533, ISBN 80-214-2890-2

[CALO] Project CALO: Cognitive Assistant that Learns and Organizes, <http://www.ai.sri.com/project/CALO>

[Carletta2003] Carletta, J., Evert, S., Heid, U., Kilgour, J., Robertson, J., and Voormann, H. (2003). "The NITE XML Toolkit: flexible annotation for multi-modal language data." Behavior Research Methods, Instruments, and Computers, special issue on Measuring Behavior, 35(3), 353-363.

[Fapso2005] Michal Fapso, Petr Schwarz, Igor Szoke, Lukas Burget, Martin Karafiat, Jan Cernocky: *Search Engine for Information Retrieval from Multi-modal Records*, poster at 2nd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms, Edinburgh, July 2005

[Ferrario2004] Ferrario, R., Oltramari, A. "Towards a Computational Ontology of Mind" in A.C. Varzi, L. Vieu (eds.), *Formal Ontology in Information Systems*, Proceedings of the International Conference FOIS 2004, Torino, November 4-6, 2004, IOS Press Amsterdam 2004, pp. 287-297

[Hain2005-I] Hain, Th., Burget, L., et. al. "The Development of the AMI System for the Transcription of Speech in Meetings", Proceedings of the 2nd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms, MLMI 2005, Edinburgh, <http://groups.inf.ed.ac.uk/mlmi05/techprog/arch/T-B-2.pdf>

[Hain2005-II] Thomas Hain, Lukas Burget, John Dines, Giulia Garau, Martin Karafiat, Mike Lincoln, Iain McCowan, Darren Moore, Vincent Wan, Rolland Ordelman, and Steve Renals: *The 2005 AMI System for the Transcription of Speech in Meetings*, in Proc NIST Rich Transcription 2005 Spring Meeting Recognition Evaluation, Edinburgh, July 2005.

[Niekrasz2005] Niekrasz, J., Purver, M., et. al. "Ontology-based discourse understanding for a persistent meeting assistant." In Proceedings of the 2005 AAAI Spring Symposium on Persistent Assistants

[OWL] Web Ontology Language OWL, <http://www.w3.org/2004/OWL/>

[Pe2004] Pečiva Jan: Collaborative Virtual Environments, In: Poster at MLMI'04 workshop, Martigny, H, DIAP, 2004, p. 1-1

[Po2004] Potůček, I., Sumec, S., Španěl, M.: Participant activity detection by hands and face movement tracking in the meeting room, In: 2004 Computer Graphics International (CGI 2004), Los Alamitos, US, IEEE CS, 2004, s. 632-635, ISBN 0-7695-2717-1

[SMARTWEB] Project SmartWeb, <http://www.smartweb-project.org>

[Su2004] Sumec, S.: Multi-Camera Automatic Video Editing, In: Proceedings of ICCVG 2004, Warsaw, Poland, 2004.

[Su2005] Sumec, S., Potůček, I., Zemčík, P.: Automatic Mobile Meeting Room, In: Proceedings of 3IA'2005 International Conference in Computer Graphics and Artificial Intelligence, Limoges, FR, 2005, s. 171-177, ISBN 2-914256-07-8.

[SUMO] IEEE's Suggested Upper Merged Ontology SUMO,
<http://suo.ieee.org/SUO/SUMO>

[Szoke2005] Igor Szoke, Petr Schwarz, Pavel Matejka, Lukas Burget, Martin Karafiat, Michal Fapso, Jan Cernocky: *Comparison of Keyword Spotting Approaches for Informal Continuous Speech*, accepted to Eurospeech 2005, Lisbon, September 2005.