FP6- 506811

AMI
AUGMENTED MULTI–PARTY INTERACTION
http://www.amiproject.org/

Integrated Project
Information Society Technologies

D6.1 PRELIMINARY ARCHITECTURE & API SPECIFICATION

Due date: 31/12/2004	Submission date: 31/12/2004
Project start date:1/1/2004	Duration: 36 months
Lead Contractor: PHI, IDIAP	Revision: 1

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

D6.1 Preliminary Architecture & API Specification

Mike Flynn (IDIAP), Bram van der Wal (PHI).

Abstract: This document describes the activities and outcomes for WP6 during the first 12 months. Section 2 presents a brief description of the AMI partner technologies to be integrated, and Section 3 discusses the proposed architecture for the demonstrator, integrating all the AMI technologies.
A lengthy appendix follows, discussing all the possible pair-wise collaborations between partners which were considered. The integration ideas are presented in alphabetic order, by proposing partner, then by collaborator. However, only a practical and relevant subset of these ideas will actually be implemented, as outlined in Section 2.

Content

1 Overview

This document describes the activities and outcomes for WP6 during the first 12 months.
Section 2 presents a brief description of the AMI partner technologies to be integrated, and Section 3 discusses the proposed architecture for the demonstrator, integrating all the AMI technologies.

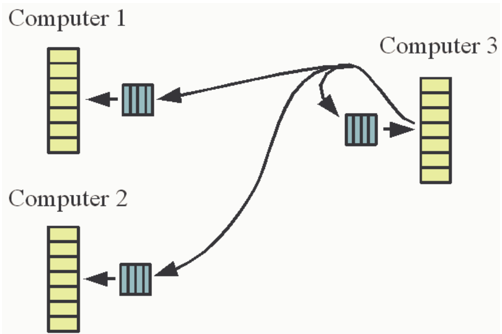
A lengthy appendix follows, discussing all the possible pair-wise collaborations between partners which were considered. The integration ideas are presented in alphabetic order, by proposing partner, then by collaborator. However, only a practical and relevant subset of these ideas will actually be implemented, as outlined in Section 2.

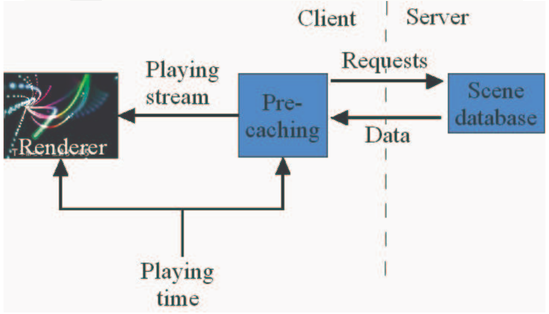
2 Partner technologies to integrate

Brno: Collaborative virtual environments (CVEs) & Keyword-spotting in speech

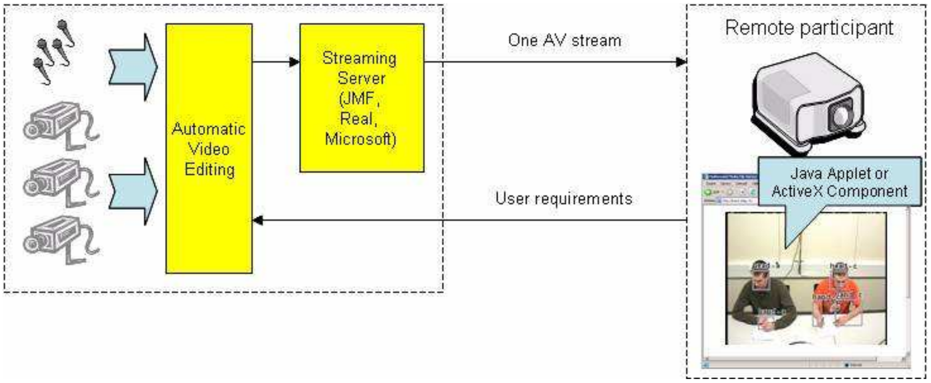
- Collaborative virtual environments (CVEs)
  - CVE system provides the meeting environment users a chance to share data, e.g. a 3D model, modify it, and discuss the data with higher quality of interaction comparing to "just video" especially when used in remote meetings.

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





- Online automatic video editing
  - Online AVE can be used to reduction of transmitted and/or displayed video streams in teleconferences. The main advantage for the user is that he does not have to watch all available video streams from remote meeting room (or more meeting rooms) but only one stream selected according to the specified requests. Other benefit is reduction of dataflow.


Automatic Video Editing
Streaming Server (JMF, Real, Microsoft)
One AV stream
User requirements
Remote participant
Java Applet or ActiveX Component

- Keyword spotting

  Keyword spotting will provide user to search the audio recordings of meetings for a specific word or combination of words. We are working on three flavours of KWS for meeting data:

  1. Based on LVCSR, which is fast, but limited by the dictionary and language model of the speech recognizer.
  2. Acoustic-based where the model of the keyword is assembled on-line after the keyword was entered - slow as it is necessary to acoustically process the entire data.
- Based on phoneme-lattices, which should combine the advantages of the previous two, e.g. fast search and possibility to search for rare/new words.

IDIAP: Browser technology

The Ferret multi-media meeting browser provides the ability to browse meetings, using a wide variety of multi-media recordings and analyses. Playback of each stream of data is synchronised, such that selecting a point in any stream, moves the playback position appropriately for all streams. New data streams are easily added and removed, dynamically. Ferret clients use Internet Explorer with RealPlayer and SVG plug-ins, together with Java Servlets and RealServer on the server side.

Philips: Wireless presentation system, from PDA

Philips CDS is highly interested in investigating how AMI technologies can be deployed in new presentation systems, providing wireless connectivity for projector systems, and how this functionality can be integrated in the AMI demonstrators. Typically, with the envisaged application, a user could spontaneously connect his mobile device to the projector (automatic network configuration) in a meeting room and then give a presentation from his mobile device. 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. In particular, this system would also enable users to give presentations from PDA's via IEEE 802.11.

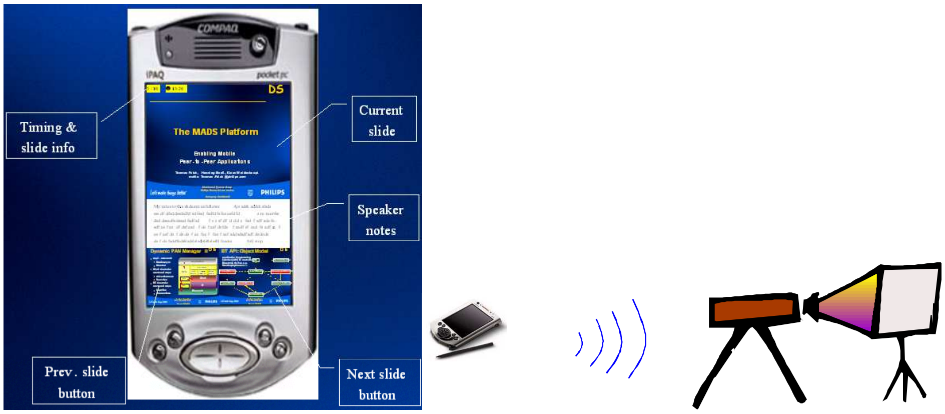


Figure 1. Wireless presenter application.

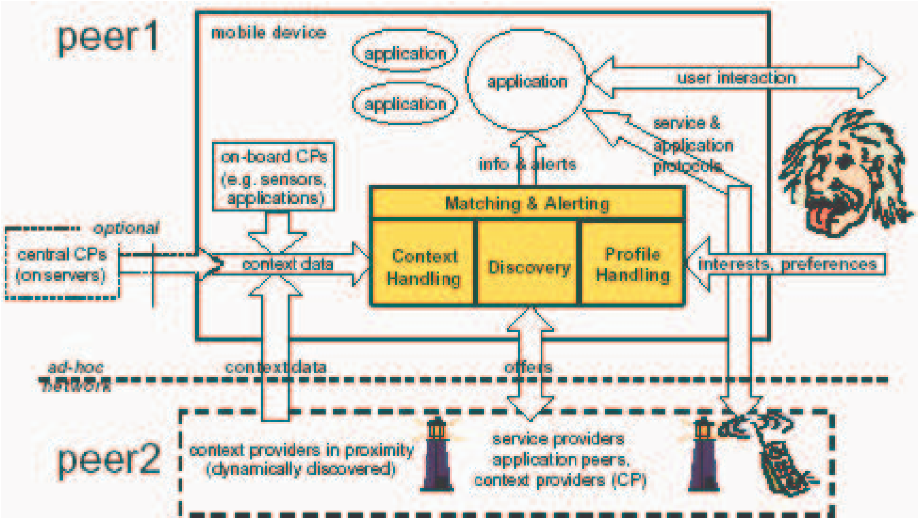


Figure 2. Core overview of the wireless presenter application.

This Wireless presenter provides a WINCE platform for mobile ad-hoc distributed systems, peer-to-peer.
For short range wireless connectivity Bluetooth or 802.11b will be used.
The following issues will be addressed:

- Establishing Bluetooth/IEEE 802.11b connections to the display device in a fast and user-friendly way.
- Define a UPnP service for exposing the display device as discoverable and accessible network resource
- Application running on the mobile device (e.g. PDA) to control the application.
- Solutions for displaying PowerPoint presentations without the need to convert them to JPEG images beforehand.

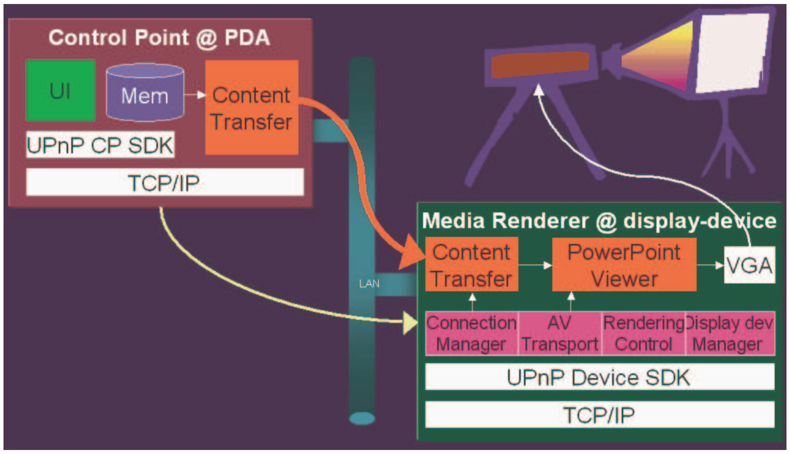


Figure 3. UPnP-based display system.

UPnP technology will be used to automatically detect, connect and control the wireless presenter application from a PDA. 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 running on the PDA, available projectors are automatically being advertised so you select the preferred projector on which to present. You are also able to chat with other PDA's with the chat function and exchange data files.

Fastcom: Development resources

Fastcom are providing design and development resources to the project – they do have a specific technology to be integrated.

RealVNC: Remote interaction tools

RealVNC is a UK company founded in 2002 by a team from the world-leading AT&T Laboratories in Cambridge. The company was established to commercially develop, enhance and promote VNC, the innovative remote access software with a widespread international user base.

VNC (Virtual Network Computing) software makes it possible to view and fully-interact with one computer from any other computer or mobile device anywhere on the Internet. VNC software is cross-platform, allowing remote control between different types of computer. For ultimate simplicity, there is even a Java viewer, so that any desktop can be controlled remotely from within a browser without having to install software.

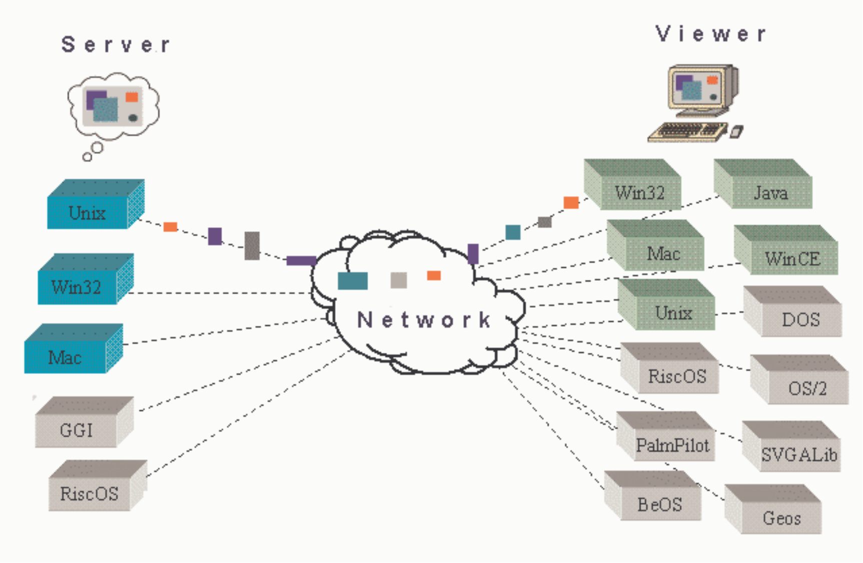


Figure 4. RealVNC architecture.

VNC has a wide range of applications including system administration, IT support and helpdesks. It can also be used to support the mobile user, both for “hot-desking” within the enterprise and also to provide remote access at home, or on the road. The system allows several connections to the same desktop, providing an invaluable tool for collaborative or shared working in the workplace or classroom. Computer support within the geographically spread family is an ever popular use.

The open source version of VNC has been freely available since 1998, and more than 20 million copies of the software have been downloaded. The software has also appeared on numerous magazine cover disks, and for several years all popular versions of Linux have included VNC. It is in active use by many millions in industry, commerce, education and at home. Virtually all Fortune 500 companies use VNC, and installations of VNC across thousands of workstations are commonplace.

Today, the RealVNC team has re-designed and re-implemented the VNC system. Compatible with the original protocols, the system is now an extremely flexible, modular and scaleable platform for vertical development, integration and licensing.

Sheffield: Semantic Compression of Audio Recordings

Speech has the advantage of being ubiquitous, expressive and easy to produce compared with text. Until fairly recently, however, there has been a lack of effective tools for accessing speech archives. A number of interface techniques have been developed in the last few years which make use of visual representations to browse and access speech. These techniques rely on there being some form of visual display; however, there remains a problem of how to access speech when there is not such a display. Semantic compression seeks to address this problem by reducing the amount of time taken to play back an audio recording whilst ensuring that the significant information can still be easily extracted from the compressed audio.

Two techniques are currently used to perform the physical compression: Audio removal and speech rate alteration. In audio removal parts of the recording are removed (for example silences and insignificant speech) reducing the length of the recording. In speech rate alteration the recording is played back faster than real-time whilst keeping the pitch each speaker constant. Note that the alteration can be applied to parts of the speech, to the entire recording or, alternatively, the degree of speed up can be continuously altered.

There are several techniques being investigated for controlling the compression. Our experiments[1] have shown that semantic approaches are more successful than acoustic approaches. Acoustic approaches analyse the waveform in order to determine the location and degree of compression. An example of an acoustic approach would be to compress parts of the recording which are spectrally similar to silence where the degree of similarity can be used to control the degree of compression. Semantic compression makes use of properties of the underlying speech to control the compression rate. For example semantic compression could compress utterances which carry less significance, according to text-retrieval measures.

Currently we have conducted a preliminary study which showed that semantic compression was more effective than acoustic compression and, furthermore, that audio removal was more effective than speech rate alteration. Further experiments will rigorously assess these results and will examine the performance of audio compression on longer meetings. Our aim is to implement semantic compression techniques as a plug-in for the AMI browser.

Twente: Multi-party interaction modelling in VR

Meetings or parts of meetings are played back in a virtual meeting room. Participants or remote participants are represented by interacting embodied conversational agents (ECAs) that simulate aspects of the behaviour of the meeting participants (e.g. gestures, gaze direction). Summaries of meetings or a combination of (parts from) different meetings can be simulated in VR as well. The advantage of such a VR simulation over replaying parts of recorded videos is that a more consistent (less fragmented) view may be generated, for instance because in the VR simulation the participants can be situated in the same place in the room in every meeting and/or remote participants can be added in the VR meeting room as if they were normal participants.

[1] Tucker, S., Whittaker, S., “Novel techniques for time-compressing speech: An exploratory study”, Presented at the International Conference on Acoustics, Speech and Signal Processing 2005, Philadelphia, USA, 18th-23rd March, 2005.

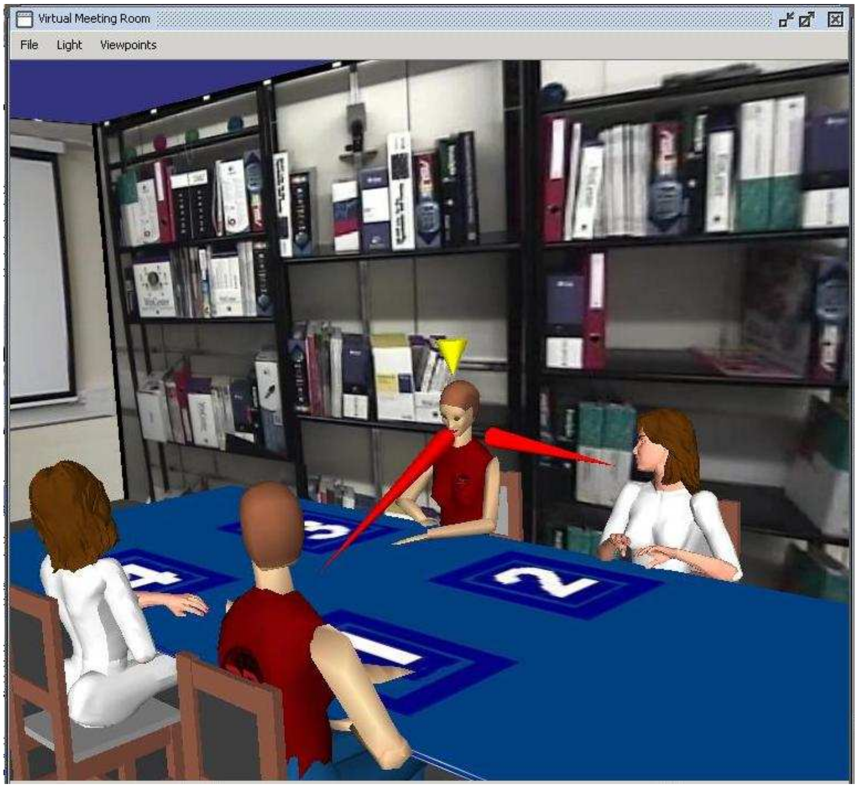


Figure 5. A virtual meeting in progress.

These simulations are generated from tracked behaviour, annotation data, summary data, and/or semantic representations. Proceeding from a high to a lower level, semantic representations (e.g. dialog intentions) are interpretations of dialog acts etc. Dialog acts are in turn (auto) recognized words and gestures (including head movements) that can be derived from direct tracking signals (sound, movement of joints). These semantic interpretations form the basis for virtual replay (video replay would be impossible for a semantic interpretation that is constructed on the basis of information from different meetings).

The view on the virtual scene can be changed: it is possible to get an overview of the room showing all the participants but it is possible to zoom in on one participant as well, or to see the room from the point of view of one of the participants. Furthermore visualisations in 2D might be added, for instance to visualize the gaze directions of the participants or to visualize who is talking.

This virtual meeting room is useful for post-meeting browsing, to enrich the browsing experience and communicate the interaction that took place – both for (potential) participants and for researchers. During the meeting the shared real and virtual environment allows visualization of the interaction, which will be especially important for remote participants. Real-time gesture and pose recognition will enable remote participants to see gaze direction, gestures and pose of meeting participants in VR. Moreover the inclusion of a virtual presenter or virtual presentation assistant becomes feasible. In research the VR simulation of meetings can be used for validation purposes: it allows fast visual feedback on the quality of tracking algorithms and it allows experiments to validate multiparty interaction models (e.g. gaze models).

For the moment we use our own VR browser. When the Java version of the Ferret browser is finished we will try to write a plug-in for that browser. Virtual simulation in Java and Java 3D, platform independent. Communication between modules is XML based. Internally we use a few C++ components.

Spiderphone: Web-enhanced conference call service

Spiderphone is building a web-enhanced conference call service that works with European public switched telephone networks. It helps groups of people (from two to a few hundred) hold meetings in which they can speak to each other by telephone, while viewing a web page to control the conference call, view who is speaking, and present PowerPoint slides. It also allows recording of these meetings, including the spoken audio, the active talker (based on telephone line), and PowerPoint slides. Spiderphone uses Computer telephony DSP boards, C++, Microsoft Windows, and standard Web technology optimized for Internet Explorer.

3 AMI demonstrator architecture overview

This section describes the architecture that will be used to integrate a demonstrator.

Abstract Architecture

Figure 6 below, illustrates a selection of the links between the technologies presented in the previous section. The links have been chosen so that the whole forms an integrated, cohesive project, with technically feasible and attainable goals. This 'abstract architecture' is described below, and then elaborated into a more technical description of a mechanism to achieve this, in the following sub-section. The abstract architecture envisages most of the AMI technologies working in conjunction with the 'Ferret' browser, rather than directly with each other. However, the RealVNC technology will probably only need to integrate with the Spiderphone conference-calling system (to record the screen events during a conference call), and with the Philips projection system. RealVNC might either record the projector output, or possibly transfer the Ferret output to a PDA, making use of the projector's wireless connection to the PDA. Meanwhile, the two VR-related technologies (CVE and VR-Meeting) could share an underlying VR infrastructure to work with the Ferret browser.

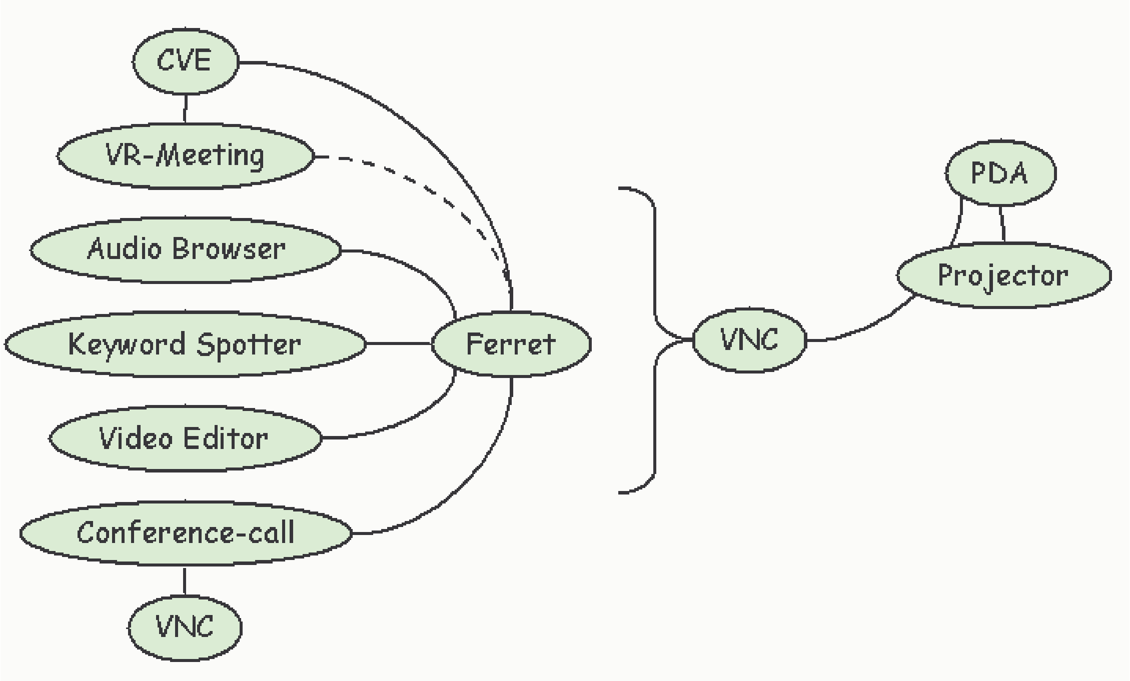


Figure 6. Abstract AMI demonstrator architecture.

The technologies to be integrated in the AMI project are technically diverse, yet all have a common goal – to facilitate the browsing of meetings. Therefore, the qualities of a technical architecture to support the project should include:

- The ability to support diverse technology (audio, video, 3D, etc.);
- The ability to run in different operating system environments (Linux/Windows);
- Flexible configuration of demonstration applications, using different combinations of technologies;
- The ability to be extended with minimal development effort, with no negative impact on currently integrated technology;
- The ability to display graphical output, controls and sophisticated text, image and media-based interfaces;
- The ability to support synchronized streaming media, from multiple sources.
- Communication and control between various technologies – so, for example, spotting a keyword in speech should then allow, say, the video to be played from the appropriate point.

Such an architecture is presented in the next subsection.

Technical Architecture

The technical architecture proposed for adoption by the AMI project is illustrated in Figure 7 below. It consists of several components, described below.

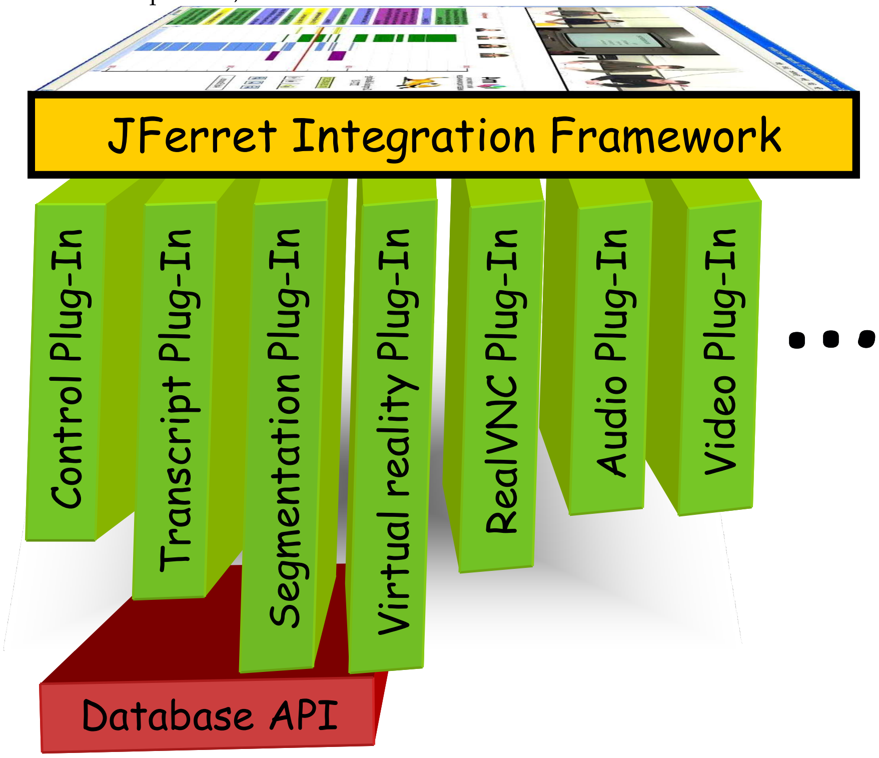


Figure 7. The Proposed JFerret Plug-in Architecture.

A relatively simple framework application, to be known as JFerret, provides a mechanism that supports 'plugins'. Plugins are software modules, which can be chosen and configured at run-time, allowing JFerret to support the diverse technologies of the AMI project. It is anticipated that JFerret, as the name implies, will be written in Java. In turn, Java supports 'late binding', allowing classes to be loaded and used at run-time, that were unknown or non-existent at compile-time.

Each of the technology-supplying partners will develop one or more plugins to be used in the JFerret framework. The framework will provide a suitable plug-in class, or interface, which plug-in developers can use as a basis for extension. In addition, any existing plug-in could be used through inheritance. Most partners are already working to some extent in Java, or can easily convert to Java. Those who cannot may still utilize the framework by communicating through sockets, files, or the Java Native Interface (JNI) – a standard mechanism allowing Java and C++ code to be linked.

Java supports multimedia through the Java Multimedia Framework (JMF). This allows multiple media sources to be played in a synchronized manner, from local files, remote files, or streaming media (except on UNIX platforms). Access to meta-data, such as speaker segmentations, will be provided through a shared database mechanism, to be determined.

JFerret will read configuration data from a file or URL, to determine which plugins to use, and to supply them with any necessary parameters. JFerret will also provide the mechanism to allow plugins to divide screen real estate to display media, graphics, text and controls.

JFerret will also supply the necessary communication and coordination mechanisms to allow plugins to co-operate, without explicit knowledge of each other. This is necessary, as plugins must be written in ignorance of the precise configuration in which they must execute. Therefore, this must be accomplished through indirect mechanisms, wherein state changes or new data are announced in an abstract manner (e.g. announcing that the 'media' is now 'playing') rather than the identity of a particular plug-in being required at compile-time. Correspondingly, plugins might register with JFerret to be told about such announcements, without knowledge of the plug-ins responsible for them.

In this manner, arbitrary plugins will combine to form an application. The coupling will be loose, through indirect (but efficient) communication mechanisms, thus allowing the plugins to come and go, throughout the execution of the JFerret framework application.

JFerret will also be responsible for managing the lifecycle of each plug-in, specifically:

- finding where the plug-in resides;
- loading the plug-in, dynamically at run-time;
- initializing the plug-in with appropriate parameters;
- sharing CPU and other resources appropriately while running;
- providing indirect communication mechanisms between plug-ins;
- stopping plug-in execution in preparation for unloading or termination;
- unloading of plug-ins, as requested by the user or the plug-in itself;
- termination of plug-ins if or when JFerret terminates.

JFerret should be capable of running both as a regular application under Windows or Linux, and as an applet – either in the 'appletviewer' or within a web-browser.

The detailed design and implementation of JFerret will be undertaken at the IDIAP Research Institute, with suitable documentation of APIs available to all partners early in 2005.

A Brno/IDIAP Integration

Description

Collaborative Virtual Environments plug-in for Ferret, shown in the figures below:
A Java plug-in for JFerret allows access to a scene database. The database contains transactions, ordered by time, along with periodic scene snapshots.

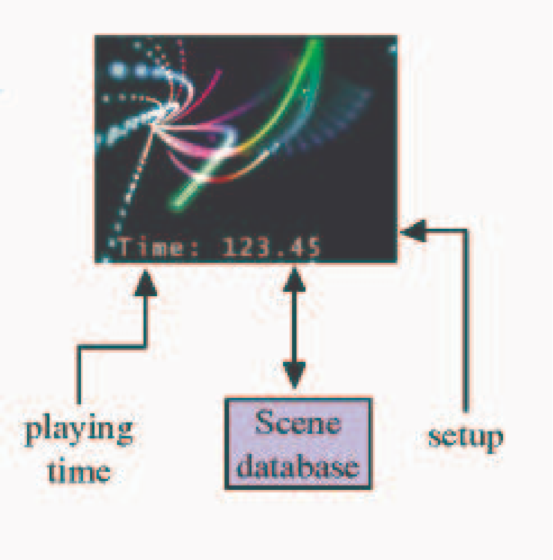


Figure 8. Brno/IDIAP Integration.

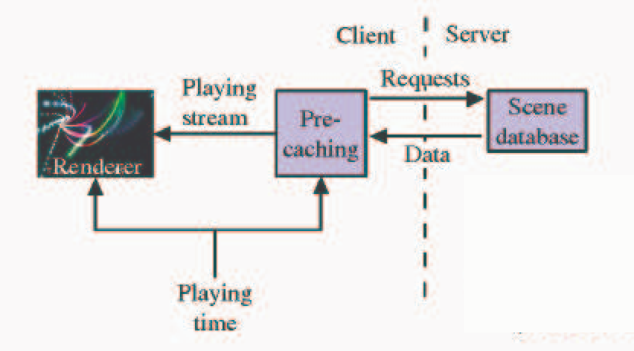


Figure 9. Architecture for CVE Playback.

Benefits to Users

In "browser" mode:

- Viewing/browsing of shared virtual scene by users;
- Export tools for virtual scene (pictures and video).

In "meeting" mode:

- Viewing of the current state of shared virtual scene;
- Manipulating/editing/interacting tools.

Technical Feasibility

Browser mode:

- Scene modifications database can be stored on a "meeting server";
- Clients are connecting to the server and they can replay/view the scene;
- Clients can export the scene or take screenshots.

Meeting mode:

- All participating computers/laptops/PDA's shares one virtual scene and all participants can interact with it;
- One computer (meeting server) stores the scene changes into the scene modifications database that can be used later for browsing.

Languages: C, C++
Operating Systems: Linux/Windows
Data:

- VRML and iv for scene import/export;
- internal format for scene modification database;
- internal format for CVE events exchanged among the computers.

Communications: TCP-IP

Research Problems

Browser mode:

- Embedding virtual scene viewer into ActiveX
- Design of scene modification database that should reside on the server
- Design of server-browser communication
- Integration into Ferret to cooperate with its components

Meeting mode:

- Network layer for communication among wires/wireless devices (probably based on sockets)
- Development of CVE consistency model and its integration into Open Inventor
- Recording changes of CVE scene for later browsing

Resources Required

From Brno:

- no additional material resources required with possible exception of C++ compiler with ActiveX support license;
- man power is assumed to fit original WP6 requirement

From IDIAP:

- co-operation with definition and implementation of Ferret interface for our ActiveX components
- co-operation on CVE plug-in into Ferret server (used while browsing through meeting record)

We believe that all the resources are available now.

Assessment

Benefits: CVE system will provide the meeting environment users a chance to share a 3D data model, modify it, and discuss the data with higher quality of interaction comparing to "just video" especially when used in remote meetings.

Costs: The cost is merely the man power planned for WP6 in Brno and a necessary co-operation cost for IDIAP.

Risks: The potential risks are the following:

- Brno will not completely succeed in implementation of all the components – this would be very bad for Brno; however, as it is relatively independent part of the project, AMI in general will probably not be harmed too much;
- Some commercial tool to perform similar task appears on the market – this option has very little probability and even if it happens, we believe to have system with good latency parameters especially suitable for the purpose.

B Brno/IDIAP Integration

Description

Online automatic video editing in teleconferences

Benefits to Users

- Transferring of single video stream automatically selected from several cameras among remote meeting participants.
- Users can specify (in terms of selection rules or directly cameras) what they want to see from the remote meeting rooms.
- Reduction of dataflow in teleconferences and possibility to use e.g. just one projector.

Technical Feasibility

- Server that produces audio video stream – on the transmission side.
- Plug-in for internet browser playing streams from servers and can control remote servers on the reception side.
- Existing technologies can be used for audio video streams transfer (Real, Microsoft) – this means virtually no additional work on the reception side.

Languages: C, C++, Prolog
Operating Systems: Windows, Linux
Data: Microsoft's Windows Media Video format, RealVideo format
Communications: TCP/IP

Research Problems

- Developing of the automatic video editing algorithm – this is mostly input from WP5 with exception of conversion of the algorithms to real-time.
- Implementation of the server using selected streaming technology – will be also used in WP5.

Resources Required

XML tags and on-line algorithms (this research does not include any actual audio/video processing but uses the data produced by WP4), e.g.:

- Tracking of meeting participants (Brno, IDIAP, Munich);
- Speaker identification (?);
- Specification of users' requirements (IDIAP, ?);
- Gesture recognition (Twente).

Assessment

Benefits: Users do not have to watch all available video streams from remote meeting room but only one stream selected according to their own requests. This can be useful because e.g. one projector can be used for projecting of the picture from the remote meeting room. Bandwidth is also saved because only one stream is transferred.

Costs: Significant part of this topic is solved in WP5. The developed algorithm has to be only extended to real-time processing and joined with streaming server. Also, on-line versions of the algorithms from WP4 (that provide the source data for the algorithm) must be developed – this is an implementation issue for WP4. The costs will probably not be directly visible in WP6.

Risks: It is possible that online versions of algorithms producing source data for the automatic video editing algorithm (being solved in WP4) will not be available or they won't be able to work in real-time.

C Brno/IDIAP Integration

Description

User interface for keyword (sound events) spotting.

Benefits to Users

The users should be able to search for the specified keywords/sounds throughout the specified portions of the meeting data.

Technical Feasibility

- Search engine (server) that searches for the keywords (sound events).
- Plug-in for internet browser capable of specifying the input data for the search engine and capable of receiving the search results from the engine.
- At this moment, we anticipate that the user interface will allow user to specify or text transcription of the keyword (sound event) to search for and/or one or more sample sounds that characterize the keyword (sound event). The sample sounds may or be predefined, or ad-hoc, e.g. one of the sounds identified "manually" in the meeting data.

Languages: C, C++, Perl
Operating Systems: Linux-server (Windows?), Windows plug-in
Data: text and audio samples
Communications: TCP/IP

Research Problems

Developing of the keyword (sound event) search – this is mostly input from WP4 with exception of the interfacing.
Development of the suitable interface for explicit specification of the sounds and also techniques, such as query by example.

Resources Required

- Meeting audio data possibly converted into phonemes or processed otherwise in order to speedup the searches (Brno).
- Sample sounds to search for, possibly special text symbols to express certain sounds (user).

Assessment

Benefits: The users should benefit from the possibility to search the data for sounds that might be not only words but also other sounds (whose nature is not known in advance) that characterize events of importance during the meetings.
Costs: Vast majority of this topic is solved in WP4. The costs in WP6 will mostly consist of implementation (and research?) of the user interface. As preliminarily discussed in Martigny workshop (Pierre Wellner and Pavel Zemcik), if the interface is simple, the interface plug-in can be integrated in Ferret by IDIAP. (So the WP6 costs will be on IDIAP side).
Risks: It is possible that the new research of keyword (sound event) spotting in WP4 will produce less than expected results. It is also possible that the interface will get too complicated and will require more research than anticipated at this moment.

D Brno/Philips Integration

Description

Handheld and Wireless slide presenter integration.

Benefits to Users

- CVE controlled from any UPnP device (PDA/laptop)
- Sending pictures/video-stream from CVE to wireless presenter

Technical Feasibility

- Port CVE software to PDA's
- PDA's should connect automatically to the meeting network and their "role" on meeting should be equal with standard computers (only for CVE), or they can serve as "intelligent remote controls" for the CVE running on a standard PC.
- Possibly some images can be captured from the scene and transmitted to a server.

Languages: C, C++
Operating Systems: Linux / MS Pocket PC 2003
Data: VRML, iv, standard picture formats
Communications: TCP-IP

Research Problems

Port of CVE software to target machines can be "adventurous" on PDA's; main portability issues should be these:

- OpenGL - or alternatives, like OpenGL ES or Mesa3D;
- Open Inventor - written in C++, very good portability over all Linux/Unices and Windows;
- SoWin - Inventor window binding library for Windows; PDA's are using MS Pocket PC 2003, so everything depends on MS Pocket PC / standard Windows API compatibility;
- Computational resources - generally less computational power than needed might be available in the PDAs.

More information about PDA's will be necessary just to say if it is possible.

CVE's consistency model limits performance to the performance of the slowest computer; because all computers have to process all CVE's network messages; Also, we may reach the limit of network bandwidth if there will be too many wireless users.

If PDAs are used as an "intelligent remote control", the tasks become much simpler and the research lies in exploiting the possibilities of controlling the CVE by the PDAs user interface (what tools can be modeled, how more than one user can interact, etc.).

Resources Required

This task was not planned in Brno before the Martigny workshop/meetings, so we did not plan any efforts in WP6. However, we might assign some of the problems as e.g. master's thesis to some students, check the feasibility, and then possibly extend the research. If some "general need" for PDA exploitation in the project exists, we can focus on this more.

Assessment

Benefits: The users can use the PDAs as PC replacements in CVE or remote controls for the CVE (depending on which approach is selected).

Costs: We did not plan any efforts in this so far. If we assign this research to the master thesis of some students, no costs (with possible exception of some SW licenses, travelling, etc.) will occur. We can reconsider this depending on the projects "general needs".

Risks: The full CVE implementation (as mentioned in the above text) is quite unexplored and generally risky in that sense those PDA resources might prove to be too limited and that various necessary software components might be unavailable.

E Brno/RealVNC Integration

Description

PDAs can be used as "remote desktop" platform for CVEs. This would require the RealVNC client/server model to be exploited for the CVE software running on the PCs.

Benefits to Users

Virtually local (remote desktop) CVE on poor performance, no-OpenGL machines, etc.

Technical Feasibility

Exploit the RealVNC remote desktop model for CVEs.
Languages: not applicable
Operating Systems: Windows/Linux-server, Windows-CE client
Data: not applicable (internal)
Communications: not applicable (internal)

Research Problems

The research will merely consist of implementation modifications of the CVE software for RealVNC architecture and in testing of the performance related issues.

Resources Required

Not very well defined at the moment. Most probably just the RealVNC software license and some PDAs not available at this moment will be needed. It is, however, possible that some extra software and/or hardware we are not aware of will be necessary (we have no experience with RealVNC).

Assessment

Benefits: The users could use their slow/low cost/mobile machines for CVE through client-server approach with (probably thick) network client
Costs: We did not plan this task before Martigny workshop/meetings and cannot anticipate the costs. It would be possible to assign this as a students project for BSc or MSc thesis and based on the results to proceed.
Risks: Not known at the moment – we have no experience with RealVNC

F IDIAP/Brno Integration

Description

Each of the three Brno technologies (CVE's, Video Editing and Keyword Spotting) could/should be accommodated within a future Ferret browser.

Benefits to Users

CVE's: when a meeting is viewed in Ferret, the virtual environment should play back, synchronized with all the other data: video, audio, transcript, slides, whiteboard and segmentations.
Video Editing: a dynamic video editor could presents the media in a simpler form, using less screen-area than multiple videos, and drawing the user's attention to interesting media streams.
Keyword Spotting: quickly find certain words in a long meeting.

Technical Feasibility

A **CVE** could be shown by an ActiveX plug-in within Internet Explorer, or a Java Applet could be used more widely.
Ferret needs to accommodate 'plug-ins' to cope with different types of data. Currently this is achieved using HTML frames, each with a certain type of data shown either by a browser plug-in or in HTML. In order to control and synchronize, the components communicate through JavaScript.
Video Editing would need to generate a video/audio stream, possibly dynamically, and feed it through the normal media display mechanism.
Keyword Spotting would probably need to be implemented as either a CGI program or servlet, together with some HTML interface. This might be simple using JSP, for example.

Research Problems

There are no research problems associated with the integration itself.

Resources Required

Brno would need to produce one (or more) of the following:

- write or find an embeddable CVE component;
- produce edited video dynamically;
- write a simple HTML interface for keyword-spotting.

IDIAP:

- clean up the mechanisms for choosing how to show data;
- clean up the control/synchronization interface.

Assessment

Benefits: One or more of the Brno technologies integrated into Ferret would show strong integration, as they would be shown alongside and co-operating with other project components.
Costs: Ferret needs to be tidied-up anyway, and the cost for that is modest. Writing a new ActiveX component for the CVE could be expensive – but one may exist, or maybe a Java Applet could be used instead?
Risks: Low.

G IDIAP/Philips Integration (a)

Description

The slides shown by the projector should be recorded and indexed for display in Ferret.

Benefits to Users

Slides are an important memorable cue into sections of the meeting recording and useful to review in real-time too.

Technical Feasibility

This should be easy, either by storing the PowerPoint slides, by recording the monitor feed to the projector, or via RealVNC technology.

Research Problems

This option does not have any associated research problems.

Resources Required

Philips would need to record the slides onto the media-server along with timing information for the database (currently a simple XML file, but later a Web-Service).

Assessment

Benefits: high
Costs:. low
Risks: low

H IDIAP/Philips Integration (b)

Description

The Philips projection server could also act as a generic meeting recording server, or proxy, with a database and media available through it.

Benefits to Users

Connection to a database and media server during a meeting, in real-time, would be simpler for users via UPnP, in the room, perhaps.

Technical Feasibility

Connection to a server for a large meeting would potentially require very large wireless bandwidth; the data may need to be physically located elsewhere, requiring a network connection; security issues.

Research Problems

This option does not have any associated research problems.

Resources Required

Philips would need to allow connections over UPnP through to the media and database servers for the meeting.

Assessment

Benefits: medium.
Costs:. medium.
Risks: medium.

I IDIAP/Philips Integration (c)

Description

A slimmed-down version of Ferret could be implemented for a PDA.

Benefits to Users

Right now, Ferret is aimed at the PC/laptop user; however, if a PDA version was available, then users do not need to bring their laptops to the meeting.

Technical Feasibility

Re-implementing Ferret components for a PDA could be very costly in time; moving to a Java-based implementation could help, provided performance was adequate; wireless bandwidth may be a problem; perhaps too difficult?

Research Problems

This option does not have any associated research problems.

Resources Required

IDIAP and others would need to port Ferret components to new languages and operating systems environments, with differing screen sizes, colour depth, audio capabilities, etc.

Assessment

Benefits: medium.
Costs:. high.
Risks: high.

J IDIAP/Sheffield Integration

Description

Variable-rate audio player plug-in for Ferret.

Benefits to Users

Listen (and watch) meetings quickly, at abnormal speeds.

Technical Feasibility

Although the technology for speeding up audio playback is well developed, there are several potential problems with browser integration:

- Changing the rate of playback, dynamically, may require switching between several possible audio tracks;
- Changing video playback rate is difficult with RealPlayer, although it is simple with the Java Media Framework (JMF) for particular rates (e.g. x0.5, x1, x2, x4 etc);
- Synchronization of other media and data streams, such as slides or whiteboard could be difficult, unless some record is maintained of the correspondence between audio-track time and original meeting time.

Research Problems

How should items of interest, in data other than the audio track, be accommodated? For example, if the slides had a lot of activity, perhaps showing an animation, we should slow down to normal speed perhaps.

Resources Required

This is difficult to determine, until Sheffield have decided upon the technology they intend to use.

Assessment

Benefits: high.
Costs:. medium.
Risks: high.

K IDIAP/Spiderphone Integration

Description

Display of remote participant identity, when speaking.

Benefits to Users

It is difficult to know who is speaking when many participants are remote.

Technical Feasibility

This is simple to achieve: Spiderphone already have information concerning who speaks when, in a conference call; adding this to Ferret is effectively just another segmentation, like any other.

Research Problems

None.

Resources Required

Access to files or real-time data, in appropriate formats.

Assessment

Benefits: high.
Costs:. low.
Risks: low.

L IDIAP/Twente Integration

Description

Use VR meeting playback in Ferret.

Benefits to Users

Unclear.

Technical Feasibility

Some kind of VR playback plug-in required (as with the IDIAP/Brno integration above).

Research Problems

Unknown.

Resources Required

Working with Brno to arrive at a standard VR display seems sensible.

Assessment

Benefits: unclear.
Costs:. medium.
Risks: low.

M Philips/Brno Integration

Description

Collaborative virtual environments.

Benefits to Users

- Control presentations and pictures from any UPnP device (PDA/laptop)
- Use multiple UPnP presentation devices (laptop/PDA) in a single meeting architecture.

Technical Feasibility

- UPnP hand-held or laptop automatically discovers meeting network.
- UPnP PDA/laptop presentation application collaborates with meeting server and UPnP projectors
- UPnP devices can share content.

Languages: Java
Operating Systems: MS Pocket PC 2003
Data: PowerPoint presentations & JPEG images stored on PDA
Communications: TCP-IP, UPnP, 802.11b wireless, Real-Time control of presentations

Research Problems

- Implement UPnP stack on Hand held
- Implement Java PowerPoint & picture viewer application on handheld.
- Design & implement UPnP PowerPoint/image rendering device

N Philips/IDIAP Integration

Description

Content sharing by Multi Input Multi Output (MIMO) concepts.

Benefits to Users

- Control presentations and pictures from any UPnP device (PDA/laptop)
- Retrieve old PowerPoint slides .
- Use multiple UPnP content devices (laptop/PDA) in a single meeting architecture.
- Use different projectors as possible display output.

Technical Feasibility

UPnP PDA/laptop presentation application collaborates with meeting server and UPnP projectors.

- Multiple UPnP devices can share content.
- UPnP hand-held automatically connects to meeting server and shares PowerPoint slides and pictures.
- Hand-held acts as a control point for the presentations.
- MIMO: Multiple PDA's / laptops can interact with multiple UPnP projectors. One UPnP node is supervisor.
- PowerPoint presentations can be stored on meeting server.

Languages: Java
Operating Systems: MS Pocket PC 2003
Data: PowerPoint presentations & JPEG images stored on PDA
Communications: TCP-IP, UPnP, 802.11b wireless, Real-Time control of presentations

Research Problems

Implement UPnP stack on Hand held
Implement Java PowerPoint & picture viewer application on handheld.
Design & implement UPnP PowerPoint/image rendering device

O Philips/Sheffield Integration

Description

Hand-held content sharing and storage on meeting server.

Benefits to Users

- Catch-up, listen again and later on view/listen the PDA's meeting content.
- Instant access to meeting server to control presentations

Technical Feasibility

- UPnP hand-held automatically connects to meeting server and shares PowerPoint slides and pictures.
- Hand-held acts as a control point for the presentations.
- PowerPoint presentations can be stored on meeting server.

Languages: Java
Operating Systems: MS Pocket PC 2003
Data: PowerPoint presentations & JPEG images stored on PDA
Communications: TCP-IP, UPnP, 802.11b wireless, Real-Time control of presentations

Research Problems

- Implement UPnP stack on Hand held
- Implement Java PowerPoint & picture viewer application on handheld.
- Design & implement UPnP PowerPoint/image rendering device

P RealVNC/Philips Integration

Integrating Wireless presenter with demonstrator architecture: The proposed change is to insert an AdderLink IP box between the PowerPoint renderer and the projector. This would allow remote users (or those who prefer to watch the presentation on their laptop) to see the slideshow in real time. It would also allow a VNC Recorder to be connected and to save the RFB stream to disk for later browsing. This is illustrated in Figure 10 below.

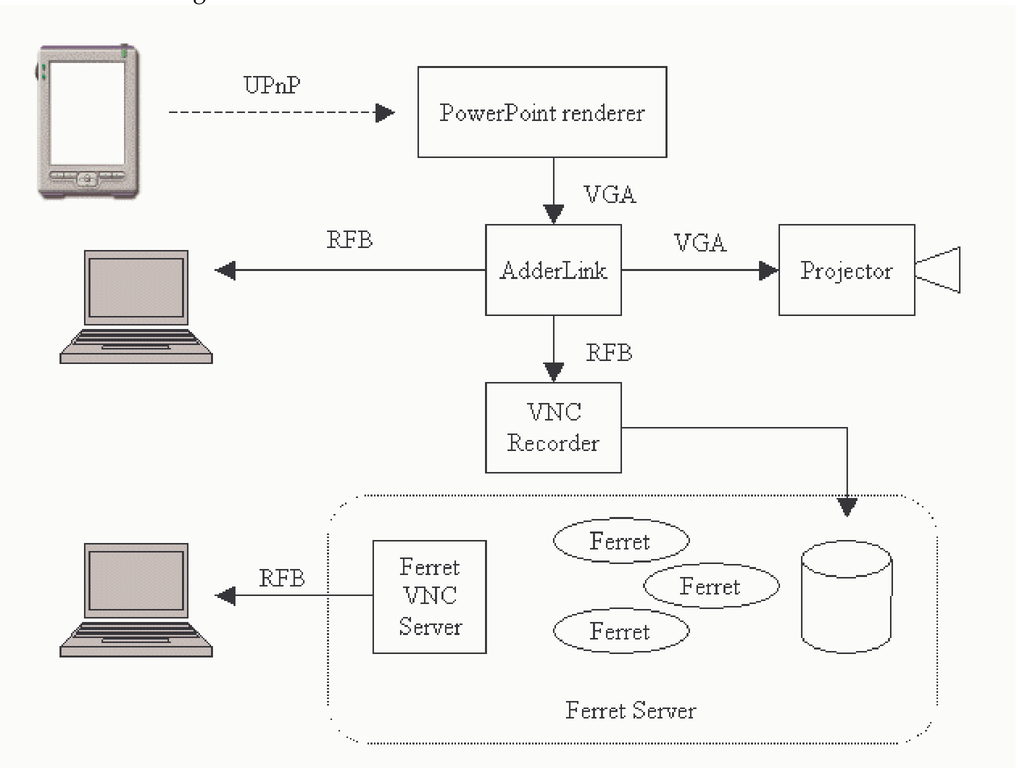


Figure 10. Wireless presenter in AMI demonstrator architecture.

All media streams are routed to these browsers (in the diagram shown above, it is done via a disk; other options, including the database are also possible). The Ferret VNC Server allows browsers to be displayed on PDAs or laptops, without requiring renderers for individual media formats to be written for each software and hardware configuration.

In the first instance, a single browser will be supported; RealVNC Ltd. will provide a modified version of VNC Server for Windows that restricts the desktop area to a single window. This approach is not suitable for multiple browsers due to technical limitations of Windows and of Real Player, and I believe the consensus was that a move to a Java-based Ferret was a good idea. This being the case, I would recommend switching to a Unix-based Ferret server; the advantages of this is that each Ferret browser would then run in a separate virtual desktop, removing issues such as control of the mouse cursor or window focus. It also has the advantage that desktops (and the Ferret browsers running in them) can be started dynamically as VNC connections are opened and closed. All of this can be provided as standard using the VNC Server for UNIX.

Regardless of the eventual choice of platform, the protocol by which participants discover the PowerPoint renderer and the AdderLink IP box should be extended to allow the Ferret VNC Server to be discovered. It will be the responsibility of this server to multiplex incoming connections to the appropriate Ferret browser/desktop.

Q Sheffield/IDIAP Integration

Description

Implementing a compressed playback component in the Ferret browser

Benefits to Users

We propose that compressed playback is a desirable feature for a meeting browser to have. This would allow users to review meetings faster than real time and to examine audio-visual meeting summaries.

Technical Feasibility

The main technical problem lies in the use of Real Media as a playback device means that client-side speedup implementations will not be possible. It has been shown[2] that server side speed up implementations are as good as client side, however our version of speed up is more involved since it must allow for both manual and automatic alterations in the speed rate and excisions which might be tricky to implement in the server.

Research Problems

Our research is focusing on the benefits of transcript-led speed up and excision. The main research problem (which we hope to address) is whether this approach is beneficial, and also what is the best implementation is.

Resources Required

The resources required will depend on the implementation path taken. If taken at the server side, then some automatic processing of the meeting data will be required. Furthermore, a component will be required at the client side to deal with choosing and synchronizing the different streams. If done at the client side then a component will be required to playback the streams and synchronize with the other components.

Assessment

The benefit of the implementation of this idea is improved usability for the user. The costs of implementing this on the client side are quite high however, since it requires an overhaul of the entire audio/video playback system. A server side implementation will be more costly in terms of storage, but the client side component will be much simpler. The risks involved in the server side implementation would be that the current a/v playback system will not have the level of synchronization required to allow changing of audio streams on the fly. Furthermore, there is the problem of synchronizing the audio playback with other browser components – though the complexity of this depends on the current implementation of this synchronization.

[2] Nosa Omoigui , Liwei He , Anoop Gupta , Jonathan Grudin , Elizabeth Sanocki, “Time-compression: systems concerns, usage, and benefits”, in *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*, p.136-143, May 15-20, 1999, Pittsburgh, Pennsylvania, United States.

R Spiderphone/IDIAP Integration Idea

Description

Spiderphone will provide tools to convert conference call recordings to a format that can be browsed using IDIAP's Ferret. The conference audio media can be made browsable (without video), and Spiderphone recordings can produce PowerPoint slides (when available) in Ferret-compatible format, as well as active speaker segmentations based on phone line activity.

As a "remote meeting assistant" is developed at IDIAP for live meeting monitoring, it may be possible for Spiderphone to feed it live active talker detection events and slide-change events. A telephone connection with acoustic echo cancelling can be included in the IDIAP smart meeting room, and this could be used to include a number of remote participants.

Another possibility is for Spiderphone to make available VoIP connections to its conference calls, to support live ASR and other types of audio processing.

Benefits to Users

This would allow browsing of meetings conducted remotely by conference call – not only meetings recorded in a specially-equipped smart meeting room.

Support from User Requirements Document TBD.

Technical Feasibility

Two main steps required:

- Copy Spiderphone recording files to IDIAP Media File Server. Authentication and access issues must be addressed both on the Spiderphone and IDIAP sides, dealing with HTTP and SSL. The process will include several manual steps at first, but could be automated if necessary.
- Convert Spiderphone data files to Ferret-compatible formats. This requires parsing of several different Spiderphone annotation files to produce Ferret-compatible XML. This could be done in a variety of languages (e.g. Java, Perl, C++) on either Windows or Linux. To make the process mostly automatic, the scripts could be launched automatically as part of the first step.

Research Problems

Ease of implementation depends in part on how well common formats have been defined for media and annotation data, and definition of these common formats is itself somewhat of a research problem. If these are not in place, however, it does not prevent this integration idea from being implemented. It simply means that custom integration is required between each partner.

Resources Required

Each partner would be responsible for providing access to its platform, and each would need to document details with examples. Either partner could take on the bulk of the conversion work. Work is simplified to the extent that each partner makes use of common AMI formats.

Assessment

This work would demonstrate the applicability of meeting processing and browsing beyond the scope of specially-instrumented meeting rooms, and it would show how AMI technology can be made accessible to people who have only a telephone and standard computer. The costs and risks are relatively low.

S Spiderphone/RealVNC Integration

Description

While participating in a web-enhanced conference call, people could show their desktops to each other.

Benefits to Users

While meeting remotely, people often need to discuss electronic documents or computer applications. If they can all be looking at the same screen/desktop they can communicate better.
Support from User Requirements Document TBD.

Technical Feasibility

It is necessary for participants to join a shared VNC session that is tied to the web-enhanced telephone conference call. To do this a shared session ID must be established and communicated among the participants. New software may need to be installed and run on the desktop that is being shared, and it may not be compatible with all platforms.
Details to be discussed.

Assessment

The screen and desktop are an important meeting resource that remote participants should be able to access like participants present in the same room.

T Spiderphone/Sheffield Integration

Description

Spiderphone conference call recordings could be converted and processed into a format compatible with Sheffield's audio meeting browser.

Benefits to Users

Spiderphone conference calls are a type of meeting that is easily recorded. Audio is the most important part of these recordings, so Sheffield's audio-only browser could easily be applied to these recordings. Support from User Requirements Document TBD.

Technical Feasibility

Spiderphone recordings would have to be converted into a format that is compatible with Sheffield's audio browser. The most efficient way to do this is for Spiderphone to convert its data into the common IDIAP Media File Server format, and for Sheffield to be able to read its data from this format as well.

Research Problems

Ease of implementation depends in part on how well common formats have been defined for media and annotation data, and definition of these common formats is itself somewhat of a research problem. If these are not in place, however, it does not prevent this integration idea from being implemented. It simply means that custom integration is required between each partner.

Resources Required

Each partner would be responsible for providing access to its platform, and each would need to document details with examples. Either partner could take on the bulk of the conversion work. Work is simplified to the extent that each partner makes use of common AMI formats.

Assessment

This work would demonstrate the applicability of meeting processing and browsing beyond the scope of video-enabled meeting rooms.

U Twente/Brno Integration

Description

We would like to investigate if it is possible to put interacting embodied conversational agents (ECAs) in collaborative virtual environments (CVEs).

Benefits to Users

See section 2 – Twente.

Technical Feasibility

- How to keep autonomous agents consistent in a CVE?
- Where should the intelligence of the ECA be located?

Research Problems

- How does an ECA behave in a multiparty virtual environment? Can it interact with more than one user?

V Twente/IDIAP Integration

Description

Replay of VR fragments using the Ferret timeline.

Benefits to Users

Users can play fragments from different meetings that deal with a certain topic or that led to a certain decision.

Technical Feasibility

- Call Java from Ferret timeline?

Resources Required

- IDIAP: Ferret;
- UT: VR browser;
- IDIAP/UT: connection between the two.

W Twente/Philips Integration

Description

Presentation of the virtual meeting room on a PDA or laptop (useful for remote participation). A virtual presenter or virtual presentation assistant on your own laptop or PDA might be useful as well.