Simple Tools for Generating a Searchable Repository of Access Grid Sessions

Ron Chernich
The University of Queensland
St Lucia, Queensland
Australia
+61 7 3365-4534
chernich@itee.uq.edu.au

Jane Hunter
The University of Queensland
St Lucia, Queensland
Australia
+61 7 3365-1092
jane@itee.uq.edu.au

Alex Davies
Australian National University
Canberra, ACT
Australia
+61 4 0079-1368
u4313969@anu.edu.au

ABSTRACT

This paper describes the VIRGIL (Virtual Meeting Archival) system which was developed to provide a simple, practical, easy-to-use method for recording, indexing and archiving large scale distributed videoconferences held over Access Grid nodes. Institutional libraries are coming under increasing pressure to support the storage, access and retrieval of such mixed-media complex digital objects in their institutional repositories. Although systems have been developed to record access grid sessions, they don't provide simple mechanisms for repository ingestion, search and retrieval; and they require the installation and understanding of complex Access Grid tools to record and replay the virtual meetings. Our system has been specifically designed to enable both: the easy construction and maintenance of an archive of Access Grid sessions by managers; and easy search and retrieval of recorded sessions by users. This is achieved through standard platform-independent web tools which don't require comprehensive knowledge or adoption of complex Access Grid tools. This paper describes the underlying architecture, tools and Web interface we developed to enable the recording, storage, search, retrieval and replay of collaborative Access Grid sessions within a Fedora repository.

Keywords

Access Grid, Videoconference, Recording, Repository.

1. INTRODUCTION

Access Grids [1, 2] have become widely established in universities and institutions globally to enable collaboration between large scale distributed teams. They support large-scale distributed meetings, collaborative work sessions, seminars, lectures, tutorials and training - group-to-group communication through an ensemble of resources that consists of multimedia displays, presentation and interactive environments, interfaces to grid middleware, and interfaces to visualization environments. Figure 1 shows the University of Queensland Access Grid Node being used for a virtual teleconference of Science and Technology Librarians.

The deployment and usage of Access Grid Nodes has grown despite the difficulties, complexities, performance problems and instability associated with the underlying IP Multicast technology and the associated Vic[3] and Rat [4] tools. Vic and Rat were originally developed by the University College London, for IT researchers to hold multi-way videoconferences over MBone [5] (multicast backbone for the Internet), and weren't designed for general use by the public. They are limited by poor robustness, packet loss, latency, low quality video and audio, imprecise lip synchronization, and the lack of a user friendly interface.

Figure 1: A UQ Access Grid Node Session

As the use of Access Grid nodes has grown, so has the demand for tools to enable the recording of Access Grid sessions so they can be replayed at a later date – by interested parties who were unable to attend the meeting or online seminar. This is of particular value when applied to online collaborative teaching sessions that consist of lectures or seminars involving multiple speakers at distributed sites.

Two previous projects have specifically developed systems to support such functionality – AGVCR [6] and Memetic[7]. Both of these systems are described in detail in Section 2.2. They provide user interfaces for recording and replaying the Vic and Rat streams separately. Memetic also enables ontology-based semantic tagging and visualization of the meeting events. However there are two critical limitations with these systems. Firstly they do not provide simple tools to enable Access Grid sessions to be recorded in platform-independent formats that can be easily replayed without the need to install Vic and Rat. Secondly they do not provide tools to enable recordings to be archived by uploading to an institutional repository (as a composite synchronized multimedia object) with associated metadata description(s). A frequently expressed request has been for a simple Web-based search interface to discover pre-recorded Access Grid sessions which can be retrieved and replayed within a Web plug-in, without the need for the installation of Vic and Rat.

This paper describes the VIRGIL (Virtual meeting Archival) system [8] which was developed to provide a simple, practical, easy-to-use method for recording, indexing and archiving large scale distributed videoconferences held over Access Grid nodes. In addition, we describe the Web search interface we developed to enable the ingest, search, retrieval and replay of the collaborative Access Grid sessions stored in the archive. Our system has been specifically designed and developed to enable managers and users to easily build, maintain and access an archive of inter-institutional Access Grid sessions without requiring detailed knowledge or deployment of complex Access Grid tools or IP Multicasting on which it is based. VIRGIL achieves this through four specific capabilities that distinguish our system from other Access Grid recording tools:

1. Sessions are recorded in formats suitable for embedding in web pages that can be played through widely available plug-ins for platform-independent Web browsers.
2. Metadata describing each session recording (and each of the embedded streams) is generated automatically. This provides the search terms for the web-based search and retrieval interface.
3. An interface is provided to upload the indexed composite digital objects to an underlying Fedora repository [9].
4. A web-based search, browse, retrieval and replay interface is provided that requires no specific software downloads.

The remainder of the paper is organized as follows:

- Section 2 describes the background information, related activities and objectives of the VIRGIL system.
- Section 3 describes the rationale behind the system's design and implementation. It also describes the functionality and user interfaces for the recording tool, database ingestion tool and search, retrieval and replay interface.
- Section 4 concludes with an evaluation of the system, a discussion of the outcomes and challenges and a list of possible future research directions and improvements to the current implementation.

2. BACKGROUND

This section provides: background information on Access Grid Nodes and their underlying technologies; a review of related activities; and the subsequent objectives and rationale behind the VIRGIL system.

2.1 Access Grid Nodes

The Access Grid is a project initiated by Argonne National Laboratories, Maths and Computer Science, Futures Laboratory in the USA [1, 2]. It is essentially an open global project to develop a large scale collaborative environment, similar to video-conferencing rooms but scaled up to allow multi-site group-to-group communication via high speed networks. There are over 200 Access Grid nodes established worldwide primarily at research universities, national laboratories, and corporate research divisions. Access Grids comprise an ensemble of resources including multimedia large-format displays, presentation and interactive environments, and interfaces to Grid middleware and visualization environments required to support group-to-group interactions across the Grid.

A typical Access Grid node is normally a room with 8-100 seats, a very large-scale display and associated computing and audio/video hardware that includes cameras, projectors, recorders, and electronic whiteboards.

Although Access Grid nodes are still being widely deployed, the user-interface to the supporting software is less than friendly, the protocol standards are still very basic, and their overall robustness is suspect. In addition, most access grid nodes require one or more dedicated operational staff to help users set up and maintain communication and tolerable audio/video quality throughout a session.

Access Grids use IP multicast for the underlying network transport protocol. Unfortunately, multicast is not always easy to deploy or to debug. Various solutions exist for "tunneling" the traffic over normal links, but they are neither scalable nor user-friendly at either end of the tunnel [9].

Access Grid software is almost exclusively open source multi-platform software. In particular it revolves around two pieces of software:

- Vic [3] the video conferencing tool, which is intended to link multiple sites with multiple simultaneous video streams over a multicast infrastructure. It uses the H.261 video codec
- Rat [4] the robust audio tool, which allows multiple users to engage in a audio conference over the Internet in multicast mode.

Vic and Rat were developed as part of the Internet Multicast backbone, or MBone [5], which provided multicast services over the unicast Internet backbone. They were designed for use by collaborating researchers. They were not designed for use by the general public, who require robustness, minimal packet loss, low latency, high quality video and audio, precise synchronization and streamlined user friendly interfaces.

Despite this, there is an increasing demand to record access grid sessions, so they can be retrieved and replayed by users – many of whom are unfamiliar with Access Grid technologies. For example, Access Grid Nodes are frequently used for seminars, tutorials and lectures which involve speakers from multiple sites. It is envisaged that the recordings of such sessions should be able to be uploaded into institutional repositories (such as Fedora[9] or DSpace [10]), described using (largely automatically generated) metadata, and then discovered, retrieved and replayed by users with little or no knowledge of Access Grid technologies.

2.2 Related Activities and Previous Work

A number of research groups have developed tools in the past for recording access grid sessions.

AGVCR [6] is a relatively mature, well-designed and easy-to-use tool for recording Access Grid sessions. It was written by Derek Piper at the Indiana University School of Informatics. Recorded files can be edited via a built-in editor. AGVCR records RTP and RTCP from multiple unicast or multicast streams (e.g. Access Grid Audio and Video) and provides the ability to replay the conference to multicast or unicast addresses. Replayed conferences are almost indistinguishable from a live session. Alternatively playback can be to a localhost by using Vic and Rat in a standalone manner from the AG toolkit. AGVCR is written in C with GTK and compiles under Linux, Windows and Mac OS.

The Voyager Multimedia Server [11] was developed at the Argonne National Laboratory to provide a scalable multi-stream record and playback engine for recording and retrieving collaborative MBone sessions. It has subsequently been extended to support the recording and playback of both Access Grid and Virtual Reality sessions such as the CAVE[12].

The MBone VCR on Demand Service [13] is a Java application that enables recording and playback of MBone sessions and the associated Vic and Rat streams. It consists of three components:

- the MVoD Server, which deals with the user management and the session management,
- the MVoD Client, which offers the users a graphical user-interface to access the MVoD Service and
- the RTP DataPump, which is responsible for the recording and playback, the synchronization and the administration of the RTP data streams.

MBone VCR doesn't provide a search interface to recorded sessions – but a Session Announcement protocol is available for announcing the availability of new sessions.

Memetic [7] is a more recent development from the University of Southampton, that began in 2005. It focuses on the capture and replay of Access Grid sessions, but with enhanced annotation functionality – primarily manual collaborative annotation tools which allow participants to create 'nodes' that record notes, issues, ideas, decisions or links to documents or websites associated with the events within a meeting. These nodes form a collaborative graphical timeline and concept map of the session that can be shared to enable semantic navigation and easy identification of relevant segments. Memetic is an extension of the Access Grid tools developed within the CoAKTinG [14] (Collaborative Advanced Knowledge Technologies) project.

All of these prior systems rely on recording and replay of separate Vic (video) and Rat (audio) streams. There may be up to 80 video streams with a single composite audio stream. Access Grid "streams" are also more complex than normal audio/video archives, in that they contain many streams, from multiple sources, and may potentially carry additional application data streams, such as slide show events and whiteboard events. The existing *rtpdump* approach does not scale well, often drops packets and does not record all of the potential material.

In addition, precise synchronization of these multiple streams at playback time is extremely difficult so it is frequently a challenge to determine who is speaking at any one time, due to poor lip-synchronization.

None of the previous systems enables recording in platform independent easily-accessible formats. None provide the ability to upload sessions to standard institutional repositories (e.g., Fedora) or provide a Web interface to search across the metadata descriptions to discover, retrieve and replay relevant sessions. Apart from *Memetic* (which relies on the manual attachment of semantic annotations), current tools only support the recording and non-interactive playback of entire Access Grid streams – our aim is to provide a tool to support richer, more interactive and fine-grained, discovery and navigation of pre-recorded access grid sessions based on the automatically generated metadata.

2.3 Objectives of VIRGIL

The objectives of the VIRGIL project were to develop a robust, efficient and interoperable system which provides:

- An easy-to-use utility (based on the VCR paradigm) for near-to-real-time recording, synchronization and multiplexing of the multiple video streams and audio stream associated with an access grid session in a format which provides: TV quality audio and video; good synchronization; low latency; and can be replayed via widely available Web plug-ins for desktop environments;
- Automatic generation of high quality, fine-grained precise metadata descriptions of the captured access grid session streams at the time of recording. The metadata should take advantage of associated session documents (agenda, minutes etc.) and should comply with pre-existing XML-based standards where available;
- Interactive editing and augmentation of metadata descriptions;
- Uploading of the session recording and associated metadata into a database or institutional repository;
- A sophisticated web-based search, browse and retrieval interface based on the underlying metadata schema;
- Presentation of selected results as dynamic HTML with an embedded link to the "movie" file allowing a user to view it in their browser;
- An interactive replay and navigation interface that provides simple easy-to-use VCR-like operations (play, pause, stop, fast-forward, rewind, time-search etc) which does not require the installation of Vic and Rat;
- Platform independence – recorded sessions should be able to be played back on any platform, regardless of what platform they were recorded on.

3. DESIGN AND IMPLEMENTATION

The VIRGIL system comprises three main components:

1. The Access Grid Recording tool – Virgil Video Recorder (VVR). This utilizes: modified versions of Vic and Rat for stream recording; the open source tool "ffmpeg" for audio/video stream aggregation and format conversion; and a user selectable utility, "mplayer" for movie playback.
2. The Metadata Editor and Repository Ingest tool.
3. The Search, Browse, Retrieval and Replay interface.

The key design challenges were to:

- Leverage existing technologies and metadata standards where possible;
- Minimize the footprint of changes to existing tools/technologies;
- Maximize portability and flexibility – design the system so that alternative optimum system components can be incorporated at runtime through easy coupling/decoupling;

- Improve the usability, user-friendliness, efficiency and presentation quality over existing systems;
- Maintain a host-neutral design with minimal and cleanly delineated host-specific code.

The next three sub-sections describe the three main architectural components of the system in detail.

3.1 The Virgil Video Recorder (VVR)

VVR differs from other tools such as AGVCR [6] in two significant ways:

1. It generates simple output in "movie" file formats (.mov and .avi) suitable for embedding in web pages that can be played through plug-ins for platform independent web browsers.
2. It generates metadata for the recording suitable for use in searchable metadata repositories that can reference the "movie" as dynamic HTML.

The VVR tool provides a rich, portable GUI environment. Written in Perl and PerlTK for portability, it interacts with modified versions of Vic and Rat used by the Access Grid Toolkit using socket based Inter-Process Communication. These tools were modified for use by the AG graphical user interface tool so that it is able to launch, manage, and close Vic and Rat sessions. The use of "hacked" versions of Vic and Rat is not ideal, but significant effort has gone into making the extent of the modifications and the process of implementing these modifications as simple as possible. Figure 2 shows the user interface to VVR.

VVR has two "tab" pages that separate the configuration and operational aspects. The settings tab is used to define default settings which are saved to a configuration file in the environment's HOME directory. Vic and Rat read this file when they start up in order to find the initial VVR port to use.

VVR controls Vic and Rat through inter-process communication (IPC) based on the passive file transfer protocol model for communications port exchange. When they initialize, Vic and Rat read a user defined socket value from the VVR properties file. They then attempt to communicate with VVR using this port. If successful, each opens a random, system-selected socket and sends that port number to VVR. VVR is then able to send commands individually to the two utilities and query them for status, record start and stop times, and stream metadata using a simple text based request/response protocol.

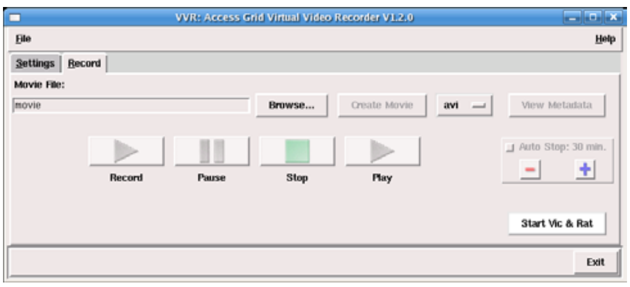

Figure 2: VVR User Interface

After starting VVR, the user may start the normal Access Grid Client software, or request VVR to launch Vic and Rat locally using URL and port values from the settings tab. Once Vic and Rat have performed their port exchange with VVR, the *record* button records the Access Grid audio and video streams separately. To minimise the real-time processing overhead, these are later multiplexed on selection of the VVR *Create Movie* button. This operation also generates XML metadata that may be edited after the event to include user supplied documentation such as the agenda, minutes etc., that will augment the envisaged search and retrieval capabilities.

The Rat utility has an existing facility to write the combined audio output to a file when given appropriate command line arguments. The modified version takes advantage of this capability by sending the file name with start/pause/finish commands via IPC from VVR.

The modified Vic utility also receives IPC commands and is responsible for extracting the metadata from un-muted video feeds. While recording, it composes a "tiled" video image in real-time from all the un-muted video streams being received. Each video stream window is labelled with the most appropriate name extracted from the stream metadata. The composite window matrix is labelled with a dynamic date/time stamp, plus an elapsed time counter to assist viewers with the location of sections of interest.

The overall movie frame size is fixed, so as more video streams join the selected venue, the individual images can be resized and fitted into the best-fit matrix. Should a feed disappear, or be muted via the Vic GUI, the space occupied may be blanked, or the matrix and image sizes re-computed. The action is specified by the user through a VVR checkbox choice.

Clicking the *Pause* button causes VVR to enter 'Record - Paused' mode. While in this mode, the network connections will be read and packets parsed for participant information, but nothing will be written to the files. Selecting *Record* or *Pause* a second time, will restart the recording of data.

When the *Stop* button is pressed, the audio and video output files are closed in preparation for post-recording processing and conversion to the selected movie format. The user must provide a filename to store the "processed" output (ie, combined audio and video streams). The same file name but with an "xml" extension is used to store the session's metadata. The raw audio and video files may be retained, or automatically deleted when VVR terminates as specified by a checkbox item on the Settings page.

Like a physical VCR, the Virgil VVR may be "programmed" to automatically terminate recording after a specific time period elapses. Post-recording processing however still requires user interaction and input.

The *Play* button can be used to open, replay and check a recorded file. Figure 3 illustrates the tiled window that displays the multiple video streams associated with an Access Grid session involving participants at 4 locations. Users are also able to fast-forward or rewind through the pre-recorded session.

Figure 3: Tiled Access Grid Session Recording

3.2 The Metadata Editor and Repository Ingest Tool

The aim of VIRGIL is to provide university librarians or collections managers with a simple set of tools to streamline the storage of new forms of complex multimedia learning objects (i.e., Access Grid recordings) within institutional repositories or digital libraries.

In order to do this, tools are required to enable the metadata automatically generated by the VVR tool to be checked for quality control, edited and augmented where necessary prior to uploading the recording and associated metadata to the institutional repository (i.e., a Fedora repository).

The metadata captured during recording is confined to that provided with the audio and video streams that identifies the participants, plus data that can be derived from the environment such as date, time and duration. To facilitate archiving and later search and retrieve operations, a step that allows additional data to be entered by a cataloguer is inserted ahead of the final storage process following upload to the repository. This metadata is arbitrary. In the prototype, we have provided elements for title, subject, agenda, and minutes of the conference. All are optional.

Figure 4 illustrates the Web-based user interface (written in PHP) that allows the user to upload new recordings with associated metadata into the chosen repository. While *http* is not the most efficient protocol for uploading large files such as the associated session recording, it is feasible for the anticipated size of recordings and is the most user-friendly option.

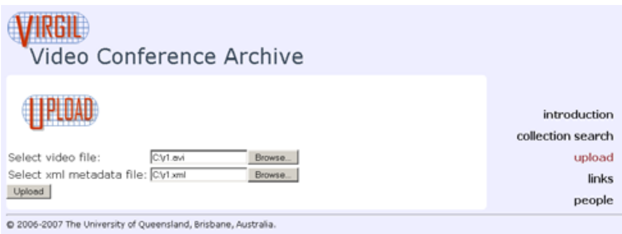

Figure 4: Session ingest tool

After the user selects the recording and associated metadata files for upload, the server parses and displays the metadata fields in an online form for review and editing.

Figure 5 illustrates the online metadata editing and input interface.

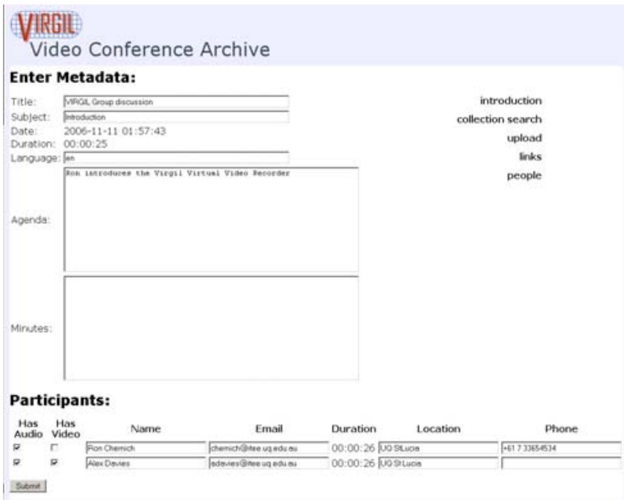

Figure 5: Metadata Editing Tool

The Metadata Editing/Input form is generated from the underlying XML-based metadata schema which was developed following a review of related efforts which included a survey report by the Terena TF-Netcast taskforce on metadata models for video-on-demand assets in academic communities (in particular videoconferences) [15, 16]. Figure 6 illustrates the structure and syntax of the metadata schema used to describe access grid sessions through an example of a metadata instance.

The metadata values in the majority of the fields have been automatically generated by the VVR tool – those that haven't been need to be manually input by the person responsible for uploading the recording to the repository.

The *participants* section of the metadata contains elements derived from the Vic and Rat metadata received with the streams. The VVR program employs a heuristic that attempts to aggregate the audio (A) and video (V) stream data originating from an individual Access Grid node. The stream attribute of the participant elements indicates whether the source is audio only, video only, or combined *AV*. This latter condition is detected by searching for matching user-name@IP-address element values in the separate Vic and Rat generated metadata.

This aggregation process is not straight forward. A session participant may not be sending video, or the session host computer may be *dual-homed* (ie, have two network cards). This is not uncommon. It is employed in large Access Grid rooms to increase the effective stream bandwidth by streaming the audio and video data over separate TCP/IP connections. In this case, the VVR heuristic will be unable to aggregate the AV metadata reliably, so the Access Grid node will appear as two separate participants. A skilled VVR operator could use the Metadata Editor to edit the metadata so as to combine *participant* instances known to originate from the same location before uploading to the repository.


```
<session>
 <title>Virgil session 25031</title>
 <date>2007-00-11T06:29:14Z</date>
 <duration>00:01:03</duration>
 <type>meeting</type>
 <description>Weekly DART project team
meeting
 </description>
 <subject>DART project progress</subject>
<rights>DART_project_permissions.rtf</rights>
 <identifier> home/xchernich/Virgil/sam3.mov
 </identifier>
 <language>en</language>
 <agenda>DART_agenda_25Oct_06.rtf</agenda>
 <minutes>DART_minutes_25Oct06.rtf</minutes>
 <recorder format="mov">
   <version>vvr.pl v1.2.0</version>
   <host.name>acer.itee.uq.edu.au
   </host.name>
   <host.address>130.102.65.240
   </host.address>
 </recorder>
 <venue>
   <rat>233.2.178.9/17008</rat>
   <vic>233.2.178.9/17018</vic>
 </venue>
 <participants>
   <participant streams="V">
     <identifier>ag@130.102.78.142
     </identifier>
     <name>Stuart Cross</name>
     <video.source>
       UQ Vislab(Audience)
     </video.source>
     <email>ag@vislab.uq.edu.au</email>
     <duration>0:01:04 </duration>
     <video.tool>vic-2.8ucl-1.1.3-AG
     </video.tool>
     <audio.tool/>
     <os>Linux-2.6.17.13-smp-i686</os>
     <location/>
     <phone/>
   </participant>
   <participant streams="AV">
     <identifier>xchernich@130.102.66.54
     </identifier>
     <name>Ron Chernich</name>
     <video.source>Ron Chernich G710(x11)
     </video.source>
     <email>chernich@itee.uq.edu.au</email>
     <duration>0:01:04 </duration>
     <video.tool>vic-2.8ucl-1.1.3-AG
     </video.tool>
     <audio.tool/>
     <os>Linux-2.6.17-1.2187_FC5-i686</os>
     <location>UQ StLucia</location>
     <phone>+61 7 33654534</phone>
  </participant>
</participants>
</session>
```


Figure 6: Example of an XML Metadata Instance

Once the metadata has been appropriately edited, merged and corrected, the associated XML file is uploaded to the Fedora repository along with the recording of the session and a snap-shot image from the recording. Fedora offers the advantage of automatically archiving previous copies of the metadata so that any changes by the person uploading edits can easily be rolled-back by the repository administrator.

After a new recording is uploaded to the Fedora repository, an RSS feed announcing the availability and details of the new Session is sent to registered subscribers. This service could easily be personalized, so only the details of new Sessions on specified topics are sent to subscribers.

3.3 The Search, Browse, Retrieval and Replay Interface

The aim of the search interface is to provide a simple, efficient search interface and high speed access and retrieval of stored Access Grid sessions – by searching on available metadata. Figure 7 illustrates the "simple" search functionality. Users can search on "Topic/Subject", "Participant Name", or "Date of Recording" or combinations of these.

The search interface was implemented as a web-based PHP interface to the underlying Fedora database. The search is implemented using iTQL which is an RDF query language. This allows the database to be searched on any of the metadata fields in the underlying schema and the results are obtained and displayed in an HTML table using PHP.

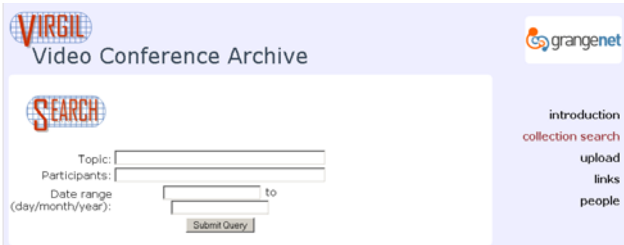

Figure 7: Simple Search interface

The Search Results page (illustrated in Figure 8) displays basic metadata - Title, Subject, Date, Duration and Participants. This is useful for further identifying the relevant sessions. The title for each retrieved result contains a hyperlink to a dynamically generated HTML page containing the full metadata description, the screen shot and a link to the multimedia composite object in the repository corresponding to that recorded Access Grid session.

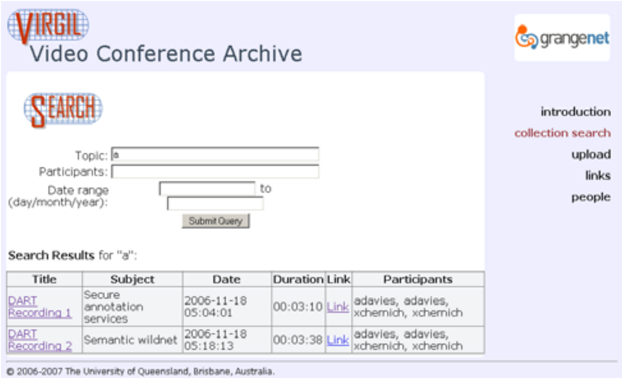

Figure 8: List of Search results

The advanced metadata page displays all of the metadata recorded for the given session, as well as a screen shot and a link to the recording. This is implemented in PHP/Fedora. Figure 9 illustrates the complete Web search results for a retrieved recording. Clicking on the screen shot opens up the plug-in which enables replay and navigation of the recording, as illustrated in Figure 3.

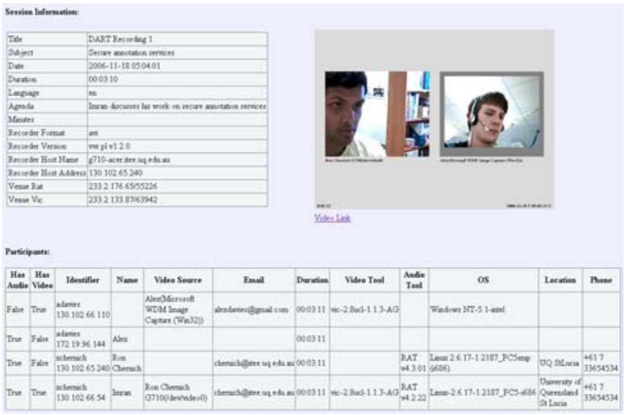

Figure 9: Session information retrieval

4. EVALUATION, FUTURE WORK AND CONCLUSIONS

4.1 Discussion and Evaluation

Usability tests were carried out by four different users selected for their lack of prior knowledge or experience of the Access Grid toolkit. The test users were first given a training session by one of the system developers which showed them how to capture a test access grid session, augment the metadata, upload it to a database and then carry out search, retrieval and replay of both the uploaded recording and other pre-recorded sessions in the archive. After the training session, the users were then asked to perform the procedure on their own. As well as being monitored during their stand-alone system testing session, they were asked for feedback at the end of their stand-alone test session.

Feedback from user testing was generally positive. However specific requests and comments from the test users led to the following improvements:

- Selected audio/video streams were able to be "muted" (ie, not appear in the movie). This request was addressed by allowing users to either blank-out or totally remove muted streams. Muting of a currently active stream, may cause a sudden rearrangement and resizing of the tiled frames. This can confuse a viewer as the location of a participant suddenly shifts in a disconcerting manner. On the other hand, simply blanking a stream may result in a sparse matrix of windows that is equally disconcerting to the viewer.
- Latency and video quality were improved by "tweaking" Vic settings. When Vic connects to a venue with a large number of video streams active, there will be a significant delay before the "full" visual matrix is built. This is due to the way that Vic conserves bandwidth by sending 8x8 pixel blocks based on a "most recently changed" algorithm. Early tests indicated that a delay of well over one minute before all block pixel groups had been sent at least once was typical for a venue with six live Vic streams. Until this is achieved, the matrix has missing blocks that adversely affect the image quality.

There were a number of additional significant challenges encountered during the development of the VIRGIL system:

- Video quality was an issue due to the way Vic minimizes bandwidth requirements. This results in very poor initial video quality that will improve over time as additional blocks of pixels are received. Reducing this initial period of poor video quality is not possible, but by participant agreement, the VVR operator can indicate verbally when the picture build-up at the recorder location has completed and recording of the session may commence to ensure that a fully formed "movie" is recorded from the start.
- Video feeds are easily added, but cannot be removed. The state of the "mute" control is encoded in the video stream. This causes a problem because a "muted" Vic video stream will cease to arrive. Hence there is no data stream to carry the new mute control state. This was addressed by trapping the Vic mute button action and sending a final "sentinel" frame that could be detected by the recorder.
- Control of "unwanted" streams is difficult as the Vic GUI is not readily monitored. The Vic modification described above provides an adequate fix for this problem.
- Control over the complex process of making the "movie" from the streams is comparatively limited. Differences in the actual start of audio and video stream recording can lead to poor "lip sync" when the two are combined in post-processing. The "ffmeg" utility used for movie creation provides ways of offsetting the start point of one component relative to the other. By using IPC, the VVR utility can obtain the individual recording start times and attempt to apply correcting factors. Experience to date shows further heuristic tailoring may be required to achieve more "natural" lip synchronization. This requires low level skill from the user in configuring the tools. Further research is required to fully optimize lip synchronization.

- Modification of Vic and Rat source code requires advanced C/C++ ability on the part of the user. To reduce the complexity of this step, an installation script was written in Perl that identifies the locations of changes in the standard distributions of the Vic and Rat source code and build scripts are generated. If the script is able to locate all points with confidence, the changes are made automatically and the results checked. If unverifiable, all changes are removed and the user must attempt manual modification from the supplied documentation.
- Significant effort was spent minimizing the footprint of the modifications (the number of changes to Rat and Vic). All the required IPC functionality was written into a common C++ base class that is extended for Vic and Rat. Modification of the two utilities requires a one line reference to the derived class that will initiate an exchange of port numbers with the VVR utility. In the case of Rat, the derived class is then able to initiate all the required control for the recording process through calls to the existing Rat functions. Unlike Rat, Vic has no native ability to write aggregate video data to a disk file. This capability is provided by an additional module that is referenced from the derived IPC controller object and fed data by a one line insertion into the Vic codebase. Mute button control requires an additional conditional statement to be inserted into the Vic source, making the overall source level changes required extremely small and simple.
- Difficulties were experienced with the calling conventions in the Rat codebase. Vic uses C++ throughout whilst Rat uses a combination of C and C++. This unpleasant surprise was discovered after the C++ base class for IPC had been written. It required some additional gymnastics in the Rat codebase and distributed Rat "makefile" script.

In addition, feedback from test users on the search and retrieval interface, led to slight modifications and extensions to the metadata schema. Two new metadata fields were added: a "type" field and a "rights" field. The "type" field is a pull-down list of access grid session types including: *meeting, workshop, conference, seminar, lecture, tutorial, discussion*. The "rights" field points to the scanned and signed permissions forms, granting permission from the participants for the session to be recorded, archived and made available either to the public or a specified user group.

4.2 Future Work

The current system could be further improved and enhanced by applying additional effort to the following issues:

- Currently we only consider the recording, description, synchronization and replay of video and audio streams. Access Grids often include other shared application events such as shared browsers, chat, whiteboards or visualizations. These data streams also need to be identified, recorded, indexed, displayed and replayed in synchronization with the audiovisual streams;
- Temporal alignment of the agenda and minutes with segments of the recorded session would enable much more precise, fine-grained search and retrieval;
- Scope exists for fine-tuning the audio and video post-recording processing to improve lip synchronization;
- The VVR recorder together with the Vic and Rat modifications were designed for cross-platform portability. However at this time, they have only been validated under Linux. For wider use, they should be validated on Microsoft Windows and Apple Mac environments. Note that while the recording must currently be made under Linux, the session participants can use standard Access Grid toolkit installations on any supported platform.
- Although the recorder tool is simple to use, building the modified versions of the Vic and Rat utilities requires skills at source code compiling. Opportunity exists to move the project to the next level by creating Install Wizards with pre-built and tested binary code for the popular target platforms.

4.3 Conclusions

This paper describes a system we have developed to enable collections managers with little or no knowledge of Access Grid technologies to quickly and easily build an archive of recordings of such collaborative virtual meetings. VIRGIL has achieved all of the objectives that were listed in Section 2.3. More specifically it enables users to:

- Record and combine all of the audio and video streams associated with an Access Grid session into a single file in a de facto format (.avi and .mov);
- Automatically generate and validate fine-grained precise metadata (conformant with an underlying XML Schema);
- Replay the recordings and edit both the recording and associated metadata descriptions for quality control purposes;
- Augment the metadata before uploading the recording to a searchable (Fedora) repository.

In addition, we have developed a Web-based search, browse, retrieval and replay interface that enables users (with no experience in Access Grid tools) to discover and replay relevant recordings stored in the repository. Usability testing has shown the system to be fast, efficient, user-friendly and the audio/video streams of reasonable quality (the best achievable within the limitations of Vic and Rat). VIRGIL provides a simple easy-to-use platform-independent system to ensure that these new kinds of complex learning objects, which capture valuable group knowledge expressed within distributed meetings, can be seamlessly incorporated in standard institutional repositories and hence are preserved for future sharing, re-use and analysis.

5. ACKNOWLEDGMENTS

This project was funded by GrangeNet (Grid and Next Generation Network), the Australian Broadband Research Network under the Australian Government's BITS Advanced Network program.

6. REFERENCES

[1] L. Childers, T. Disz, R. Olson, M. E. Papka, R. Stevens, and T. Udeshi, "Access Grid: Immersive Group-to-

Group Collaborative Visualization," *Proc. 4th International Immersive Projection Technology Workshop*, 2000.
[2] "Access Grid Homepage," www.accessgrid.org.
[3] S. McCanne and V. Jacobson, "vic: A Flexible Framework for Packet Video," *ACM Multimedia*, vol. 95, pp. 511-522, 1995.
[4] V. Hardman, A. Sasse, M. Handley, and A. Watson, "Reliable Audio for Use over the Internet," in *INET'95*, Honolulu, Hawaii 1995.
[5] H. Eriksson, "MBONE: the multicast backbone," *Communications of the ACM*, vol. 37, pp. 54-60, 1994.
[6] D. Piper, "AccessGrid Video (Cassette) Recorder." http://iri.informatics.indiana.edu/~dcpiper/agvcr/.
[7] S. Buckingham Shum, R. Slack, M. Daw, B. Juby, A. Rowley, M. Bachler, C. Mancini, D. Michaelides, R. Procter, and D. De Roure, "Memetic: An Infrastructure for Meeting Memory," *Proceedings 7 thInternational Conference on the Design of Cooperative Systems, Carrey-le-Rouet, France*, 2006.
[8] VIRGIL project Home Page, http://www.itee.uq.edu.au/~eresearch/projects/virgil/index.html.
[9] "Fedora," www.fedora.info.
[10] "DSpace," www.dspace.org.
[11] T. Disz, I. Judson, R. Olson, and R. Stevens, "The Argonne Voyager multimedia server," *High Performance Distributed Computing, 1997. Proceedings. The Sixth IEEE International Symposium on*, pp. 71-80, 1997.
[12] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti, "Surround-screen projection-based virtual reality: the design and implementation of the CAVE," *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pp. 135-142, 1993.
[13] W. Holfelder, "MBone VCR: video conference recording on the MBone," *Proceedings of the third ACM international conference on Multimedia*, pp. 237-238, 1995.
[14] CoAKTinG Project, http://www.aktors.org/coakting/.
[15] H. K. Salminen, "Metadata survey report," April 2003 http://www.terena.org/activities/tf-netcast/deliverables/deliverable-a.html.
[16] H. K. Salminen, "TF-netcast: Final Report" , September 2004 http://www.terena.org/activities/tf-vvc/docs/Deliverable_I.pdf