VIRGIL – Providing Institutional Access to a Repository of Access Grid Sessions

Ron Chernich, Jane Hunter and Alex Davies

The University of Queensland, St Lucia, Queensland, Australia
{chernich, jane}@itee.uq.edu.au
&
Australian National University, Canberra, ACT, Australia
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 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.

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 scalable group to group (G2G) communication over network connections. 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.

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. 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. 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).

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. 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.

2. Background

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 G2G communication via high speed networks. There are over 200 Access Grid nodes established worldwide primarily at research universities, national laboratories, and corporate research divisions.

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. 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.
- 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 unfamiliar with Access Grid technologies. The recordings of such sessions should be able to be uploaded into institutional repositories (such as Fedora [9] or DSpace [10]), described using 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. AGVCR records RTP and RTCP from multiple unicast or multicast streams 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.

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. MBone VCR doesn't provide a search interface to recorded 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. 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. 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 formats. None provide the ability to upload sessions to standard institutional repositories, nor 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 that can be replayed via widely available Web plug-ins for desktop environments;
- Automatic generation of high quality, fine-grained metadata;
- Interactive editing and augmentation of metadata descriptions;
- Uploading of the session and associated metadata into an 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" providing platform independent replay;
- An interactive replay which does not require the installation of Vic and Rat.

3. Design and Implementation

The VIRGIL system comprises three main components:

1. The Access Grid Recording tool – Virgil Video Recorder (VVR).
2. The Metadata Editor and Repository Ingest tool.
3. The Search, Browse, Retrieval and Replay interface.

The key design challenges were to leverage the existing technology to provide a user friendly, host-neutral design with a minimal change footprint.

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. 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 1 shows the user interface to VVR.

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.

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.

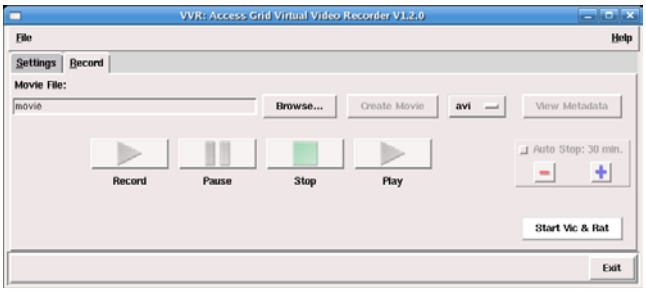

Fig. 1. VVR User Interface

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 are 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.

Recording may be paused at any time. While in this mode, the network connections will be read and packets parsed for participant information, but nothing will be written to the files. 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.

Fig. 2. Tiled Access Grid Session Recording

3.2 The Metadata Editor and Repository Ingest Tool

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.

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 [15, 16].

The metadata values in the majority of the elements are automatically generated by the VVR tool. Other may 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.

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. 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 3 illustrates the "simple" search functionality.

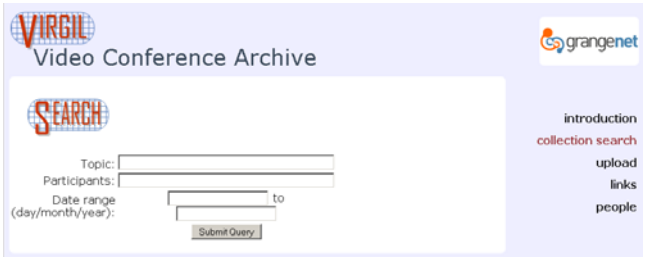

Fig. 3. Simple Search Interface

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.

The Search Results summary page 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 record.

The full 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 4 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 2.

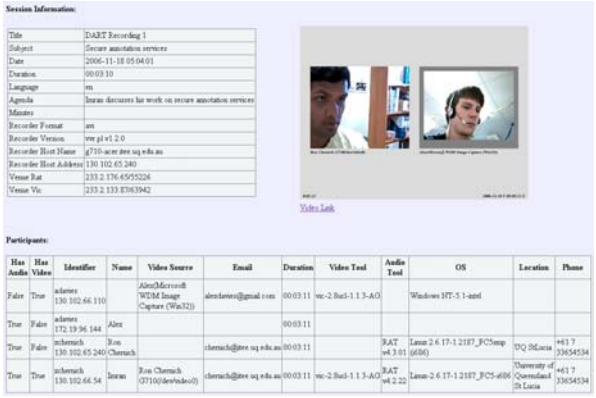

Fig. 4 Session Information Retrieval

4. Evaluation, Future Work and Conclusions

4.1 Discussion and Evaluation

Usability tests were carried out by four different groups selected for their lack of prior knowledge or experience of the Access Grid toolkit. 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.
- 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:

- 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.
- 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 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. 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.
- 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 repository.

Acknowledgements

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.

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