Vannotea – A Collaborative Video Indexing, Annotation and Discussion System For Broadband Networks

Ronald Schroeter
The University of Queensland
Brisbane, Queensland, Australia
ronalds@dstc.edu.au

Jane Hunter
DSTC
Brisbane, Queensland, Australia
jane@dstc.edu.au

Douglas Kosovic
DSTC
Brisbane, Queensland, Australia
douglask@dstc.edu.au

ABSTRACT

A number of research groups and software companies have developed digital annotation tools for textual documents, web pages, images, audio and video resources. By annotations we mean subjective comments, notes, explanations or external remarks that can be attached to a document or a selected part of a document without actually modifying the document. When a user retrieves a document, they can also download the annotations attached to it from an annotation server to view their peer's opinions and perspectives on the particular document or to add, edit or update their own annotations. The ability to do this collaboratively and in real time during group discussions is of great interest to the educational, medical, scientific, cultural, defense and media communities. But it is extremely challenging technically and demands significant bandwidth, particularly for video documents. In this paper we describe a unique prototype application developed over the Australian GrangeNet broadband research network, which combines videoconferencing over access grid nodes with collaborative, real-time sharing of an application which enables the indexing, browsing, annotation and discussion of video content between multiple groups at remote locations.

INTRODUCTION

This paper describes a unique prototype system developed at the Distributed Systems Technology Centre, at the University of Queensland, which enables the real-time collaborative indexing, browsing, description, annotation and discussion of high quality digital film or video content. Using the GrangeNet broadband research network [1] and access grid nodes [2] which support large-scale group-to-group collaboration and high quality audio/video, users are able to open an MPEG-2 file and share the tools which enable the group to collaboratively segment, browse, describe, annotate and discuss the particular film or video of interest. Although annotation tools do exist for textual documents, web pages, images, audio and video resources, they have been designed for use within stand-alone environments. The descriptions and annotations can be shared by saving them to a server, but the actual annotation applications have not been designed to be shared in real-time collaborative video-conferencing sessions. Hence, the Vannotea system is of great interest to many communities, including the educational, medical, scientific, defense and media communities, to enable collaborative online discussions about particular film or video content and real-time annotation of segments, key frames or regions within keyframes between distributed groups.

RELATED WORK

Indexing and annotation systems for digital video files have been developed in the past - but only for use within stand-alone environments in which the annotations can be saved and shared asynchronously. Our first task was to carry out a detailed survey of these existing systems, determine their best and worst features and integrate the best features in a prototype which could be shared within a collaborative real-time high-quality video-conferencing environment.

A survey of existing video annotation systems revealed that the following systems were the most advanced:

- IBM – MPEG-7 Annotation Tool [3]
- Ricoh – Movie Tool [4]
- ZGDV – VIDETO [5]
- COALA – LogCreator [6]
- CSIRO's CMWeb tools [7]
- Microsoft's MRAS [8]

IBM's **MPEG-7 Annotation Tool** provides support for both MPEG-1 and MPEG-2 files as well as regional annotations. It also comes with a shot detection algorithm, an easy-to-use interface and a customisable lexicon. However, the UI is restricted to a pre-set video size and aspect ratio. If a video has a different format than it cannot be displayed correctly. The lexicon is also restricted to three default categories (event, static scene and key objects), although free text keywords can also be added. IBM's system doesn't support hierarchical video segmentation.

Ricoh's **MovieTool** does support hierarchical segmentation within a timeline-based representation of the video. The automatic shot boundary detection algorithm permits changes to threshold settings. The MovieTool is the most mature and complete of the systems, but has a complicated user interface which is closely tied to the MPEG-7 specification. The user has to have a good knowledge of the large and complex XML Schema definition of MPEG-7 in order to browse using the MPEG-7 Editor.

In contrast, ZGDV's **VIDETO** hides the complexity of MPEG-7 basing the description properties on a simple description template, which can then be mapped to MPEG-7 using XSLT. Domain-specific description templates together with their corresponding XSLT mappings are generated. The resulting flexibility, customisability and user-friendliness of this approach are VIDETO's biggest advantages. VIDETO was developed as a research tool to generate video (XML) metadata for testing a video server and retrieval module.

The **LogCreator** of the COALA project is a web-based tool which supports video descriptions. It offers automatic shot detection and a good interface for hierarchical segmentation of videos that can be uploaded to the server, where it is saved as MPEG-7 in a native XML database. However, it is a domain-specific tool, developed specifically for TV news documents with a predefined structure. The descriptors that are used to annotate the different video segments are predefined as well.

Two other web-based video annotation systems are: CSIRO's **Continuous Media (CM) Web Browser** which generates a proprietary HTML-format Annodex file [9]; and Microsoft's Research Annotation System (**MRAS**) [8] – a Web-based application designed to enable students to asynchronously annotate web-based lecture videos and to share their annotations.

None of the systems described above are designed to be used within a collaborative video-conferencing environment. Microsoft's Distributed Tutored Video Instruction (DTVI) [10] system does allow students to replay and discuss videos of lectures collaboratively. However it does not support real-time synchronous annotations. It is also based on a combination of Windows Media Player and Microsoft's NetMeeting [11]. Net Meeting is based on the T.120 protocol [12] for application sharing. Because T.120 has been designed for low bandwidth and only supports low frame rates (e.g., 10fps), the capture and transfer of mouse events, keyboard events and screen update to the display devices of the participants is too slow to adequately handle high quality MPEG-2 video (24-30fps).

Consequently we were unable to use the NetMeeting application-sharing capabilities and had to develop our own collaborative application sharing environment from scratch using .NET Remoting. The sub-section on .NET Remoting describes this in more detail.

OBJECTIVES

An analysis of existing systems enabled us to determine the objectives of this project in more detail. Our primary goal was to develop a system to enable the collaborative indexing, browsing, annotation and discussion of video content between multiple groups at remote locations. In addition the system must support:

- User/group participation via access grid nodes;
- Delivery over the GrangeNet broadband research network;
- High quality video – MPEG-2 files;
- Automatic shot detection;
- Hierarchical video segmentation;
- Simple user interfaces;
- Flexibility – different domains, communities and metadata application profiles;
- International video metadata standards such as MPEG-7;
- Annotation of segments, shots, frames and regions within frames;
- The ability to save, browse, retrieve and share both the authorized, structured, objective metadata/descriptions as well as the subjective annotations and their source (who said what and when).

ARCHITECTURE

Figure 1 illustrates the overall system architecture – assuming deployment within an educational context. The scenario is a live discussion between students and lecturers from tertiary Film/Media Studies Departments, communicating with curators, archivists and film/media analysts from leading audiovisual archives and the creative industries in Australia - via access grid nodes over the GrangeNet broadband research network. All of the participants of this hypothetical online videoconference are sharing an application which enables the retrieval of an MPEG-2 video and real-time collaborative, synchronous indexing, browsing, annotation and discussion of the video.

Our assumption is that there are two separate metadata stores: one store is for the search and retrieval of video content from the servers (we assume that this will be provided and maintained by the custodial organization); and a separate metadata store for logging the shared personal annotations. Our distinction is based on the premise that the first one stores objective authorized descriptions of the content, provided by trained cataloguers using controlled vocabularies, whilst the

second store contains personal and highly subjective views, expressed in free text, which are clearly attributed to specific individuals rather than organizations. In the real world and within the Internet, this distinction often becomes highly fuzzy. Our software enables both types of metadata to be entered and saved.

The video content is being streamed from multiple video servers located at different custodial organizations e.g., ScreenSound Australia [13] (the Australian National Audiovisual Archive) or the Australian Centre for Moving Image [14].

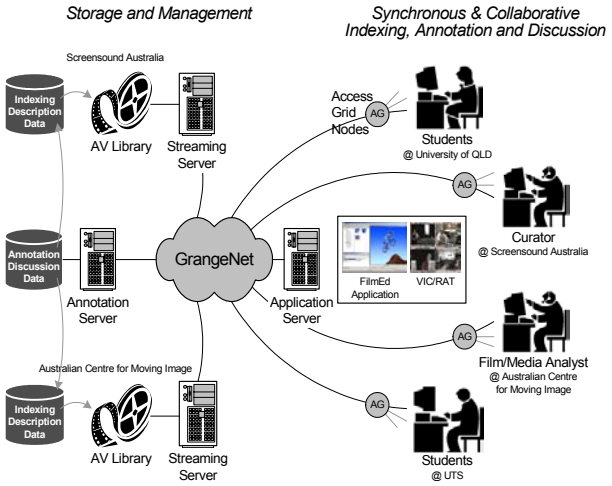

Figure 1: Overall System Architecture

COMPONENTS

The first phase of the project consisted of the development of a simple stand-alone video indexing, browsing and annotation prototype which supported the features described in Section 0. The second phase consisted of integrating this as a shared application within the collaborative videoconferencing environment. The development environment chosen was Visual Studio .NET and the C# programming language. Java Media Framework was unsuitable because of its lack of support for MPEG-2. Figure 1 illustrates the four major components of the system which needed to be developed and which are described in more detail below:

- Search and Retrieval Database;
- Annotation Database;
- Application Server;
- MPEG-2 Streaming.

Search and Retrieval Database

The first task in developing this database was to specify the underlying metadata schema(s) necessary to enable the search, retrieval and browsing of video files stored on the streaming video servers connected to the network. A simple application profile which combines Dublin Core[15] and MPEG-7 [16] was developed to enable both the resource discovery of atomic video files as well as the fine-grained retrieval of relevant video segments [17]. Figure 2 below illustrates the data model for the search, retrieval and browsing metadata. An automatic shot detection module provided by Mediaware [27] was integrated to automate the segmentation and hence the metadata generation, as much as possible.

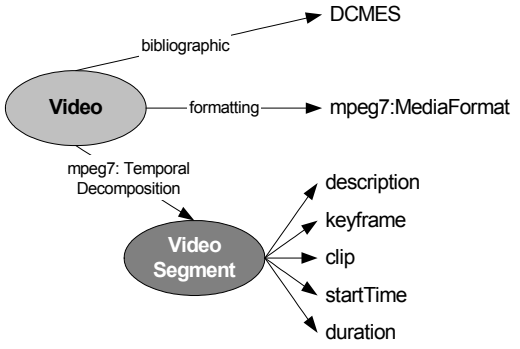

Figure 2: A Generic Descriptive Metadata Model for Moving Image

Annotation Database

The annotation database stores the annotations (which may be associated with segments, keyframes or still regions within frames), as well as the source of the annotations (who, when, where). Annotations can be notes, explanations, or other types of external subjective remarks. We decided to base the annotation component of our software on Annotea [18], an open system developed by the W3C, which enables shared annotations to be attached to any Web document or a part of the document.

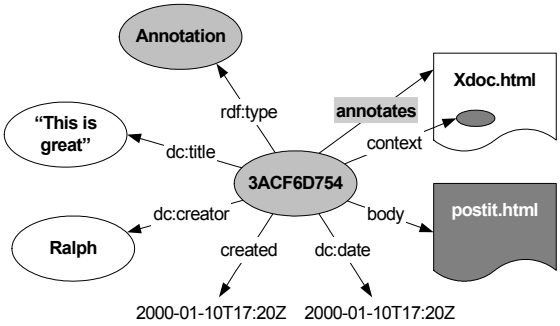

Figure 3: The basic annotation schema [21]

Annotea uses an RDF-based annotation schema [19] and XPointer [20] for linking the annotations to the document. Figure 3 illustrates the basic annotation schema employed by Annotea. We have extended this to

support the annotation of audiovisual documents - "context" is specified through extensions to XPointer which enable the location of specific segments, keyframes or regions. This approach also allows us to utilize and test prototypical annotation server implementations such as Zope [22] or the W3C Perllib [23] server. These are RDF databases which sit on top of MySQL and provide their own query language, Algae [24].

The "body" of an annotation is usually text or HTML. But our architecture allows us to generate, attach and store audiovisual annotations - small audio or video clips captured during the video conferencing discussion.

Application Server

Application Sharing Protocols

The approach adopted by application sharing protocols such as T.120 (NetMeeting) or VNC-Protocol [25] makes them unsuitable for our application. In such protocols, the shared application runs on a master client or server, which receives the keyboard and mouse events from the participants and sends captured screen/window updates back to the participants (Figure 4).

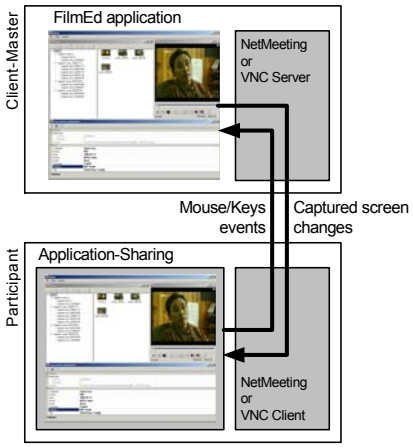

Figure 4: Application Sharing Protocols

The main advantage of this approach is that a single framework can be used to share different applications. However, these protocols were designed for low-bandwidth networks and can not handle the high frame rates required by MPEG-2. They also restrict the application sharing to a single user being in control at any one time. Because of our need to support high frame rates and MPEG-2, such ready-made application-sharing frameworks are unsuitable. We have had to build a collaborative environment from scratch, using .NET Remoting. This is described in detail in the next section.

.NET Remoting

Because the Vannotea prototype is implemented in C# within the .NET development framework, the most flexible, modular and integrated approach to application sharing was to develop it using .NET Remoting. .NET Remoting provides a framework that allows objects to interact with each other across application domains or on different servers. All of the language constructs, such as *constructors, delegates, interfaces*, *methods, properties* and *fields* can be applied to remote objects. Calling a remote object is the same as calling a local object. When combined with the mechanisms of *delegates* and *events*, remote objects can also call methods on the client. Even arguments can be passed as long as they are serializable.

Figure 5 illustrates the event-handling architecture of our application. In this example, the client-master is in control of the application, the remote clients are joining the session by connecting to the same server-application.

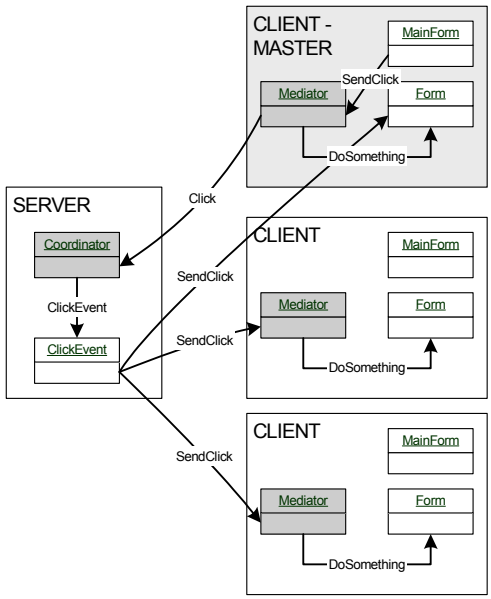

Figure 5: Event handling using .NET Remoting

The Mediator objects handle the communication between the clients and the server. They can call methods on the remote object (Coordinator). In return, the Coordinator can call methods on the Mediator by raising events that the Mediator has subscribed and listens to.

The goal is to simulate all events on all clients. In Figure 5 the client-master clicks a button in the MainForm, which is then reflected in another Form of **all** clients. After a button click, an event is raised and handled by forwarding this information to the Mediator object. The Mediator checks the information and calls a method on the server, telling it that a button has been clicked. The server then raises a ClickEvent that each

client's Mediator object has subscribed and listens to. Finally all Mediators handle the event by doing something in their Form.

Mouse movement events are handled in the same way. The client master's mouse position is updated and transferred to all clients, where it is displayed as a pseudo mouse pointer. This provides the necessary feedback to users about what the other user did and where he/she clicked.

The approach described above assumes that one user is in control at any time. Alternatively every remote client could be the client-master at once, creating a truly collaborative environment for the application in which every participant is in control simultaneously, resulting in several mouse pointers within one application. To differentiate between users, the mice would be colour-coded. Such a scenario may sound chaotic - however in certain situations, it may actually be useful to have multiple users doing different tasks synchronously.

One objective of the project is to evaluate users' behavior and obtain user feedback on the different levels of collaboration available during video analysis and discussion and annotation processes. Although the design approach which we have adopted is more difficult in the short term, over the longer term it provides the required flexibility to explore these aspects fully and easily modify the system in response to user feedback and evaluation.

Combined with the MPEG-2 streaming architecture described in the following section, this approach also fully utilizes the advanced bandwidth and low latency capabilities provided by GrangeNet.

MPEG-2 Streaming

The Server sends VCR-like commands (play, pause, seek, stop) to the Streaming Server, which then streams the section of the MPEG-2 file that needs to be played and viewed on the remote clients.

For efficiency and scalability IP Multicasting is used for the streaming. Without multicasting, the same information would need to be carried over the network multiple times, via separate unicast streams for each remote client.

The transport protocol being used for the MPEG-2 multicast streams is UDP (User Datagram Protocol), which provides end-to-end delivery services for data with real-time characteristics, such as interactive audio and video. To receive the MPEG-2 over UDP stream, the clients use a DirectShow Filter for UDP reception.

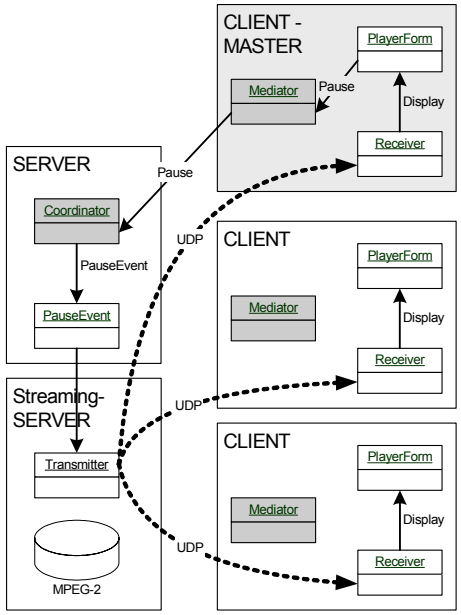

Figure 6: MPEG-2 Streaming

IMPLEMENTATION

Description Architecture

A key objective of the system was to provide simplicity and flexibility for users in their choice of metadata descriptions, whilst still supporting standards and interoperability. This required a design which could easily adapt to the different application profiles required by different communities. We did this by providing a tool which enables users to define and edit XML Description Templates – simplified versions of XML Schemas. For example, the Description Template in Figure 7, defines domain-specific hierarchical structures for "Film" and their relevant description i.e., a feature film will be segmented into scenes and shots. A Film description would typically include: Title, Creator, Genre, Date, etc. TV News on the other hand might be segmented into presentations, reports and interviews, which would require a different Description Template.

```
...
<!-- ********************************** -->
<!-- User-defined hierarchal structure    -->
<!-- ********************************** -->
<SegmentHierarchy>
 <Segment type="Film">
  <Segment type="Scene">
   <Segment type="Shot"/>
  </Segment>
 </Segment>
</SegmentHierarchy>
```

```
<!-- ********************************** -->
<!-- User-defined Description Elements    -->
<!-- ********************************** -->
<Descriptions>
 <Description type="Film">
   <DescriptionElement name="Title"/>
   <DescriptionElement name="Creator"/>
   <DescriptionElement name="Genre"/>
   <DescriptionElement name="Date"/>
 </Description>
```

Figure 7: Simplified example of a Description Template

The User Interface for entering metadata, is dynamically generated from the Description Template and reflects the segment hierarchies and description elements defined within it. The metadata for each video file is represented as a Description DOM (Figure 8) similar to the structure of the template, which makes it simple to transform to different standards like Dublin Core and MPEG-7 [17] using XSLT.

```
<Vannotea>
 <Segment type="Film" id="media_1">
  <Description type="Film">
       <DescriptionElement name="Title">
       <DescriptionElement name="Creator">
       <DescriptionElement name="Genre"/>
       <DescriptionElement name="Date"/>
   </Description>
   <Segment type="Scene" id="scene_0">
    <Description type="Scene">
     <DescriptionElement name="SceneTitle">
     <DescriptionElement name="FreeText">
    </Description>
  </Segment>
 </Segment>
</Vannotea>
```

Figure 8: Simplified example of a Description

User Interface

A full-size screen capture of the interface, being used in the context of an access grid session, is available in Appendix A. Figure 9 illustrates the three key components of the user interface:

- The *Content Player* displays the video content being streamed from the archive or custodial organization;
- The *Content Description* component enables the objective and authorized segmentation and indexing of the content, as well as search, browsing and retrieval;
- The *Annotation & Discussion* component enables the input, logging, search and retrieval of shared annotations.

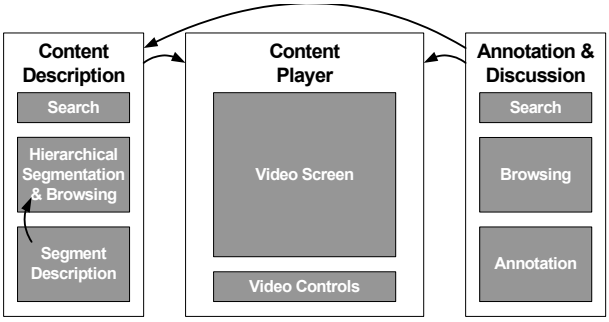

Figure 9: User Interface components

In order to streamline the indexing and segmentation process, an automatic shot-detection capability was added. The Mediaware SDK [27] is used to perform the automatic shot-detection. This generates a list of shots for the entire MPEG-2 file or a selected segment. Because the Mediaware SDK is written in C++, C# wrappers needed to be developed in order to integrate it.

Once the shot-list has been generated, the explorer-style **browser** in the **Content Description** window allows either further hierarchical segmentation of shots to frames or aggregation of shots to higher-level segments (scenes). This hierarchical structuring into: segments, scenes, shots and frames; enables easy navigation through the video.

Also within the **Content Description** window, selected segments or frames can be described either by entering free text values or using controlled vocabulary/terms available through pull-down menus.

The **Content Player** features common video playback functionalities (play, pause, seek, stop) and also allows the annotation of the current video frame, through the use of drawing tools to define regions. The drawing tools support the attachment of annotations to rectangular, point or linear regions within frames. The actual annotation for a region is input via the ***Annotation and Discussion*** window. Details of who attached the annotation and the date/time of annotation are also recorded.

The **Annotation and Discussion** window also enables users to browse and display past annotations and to see to whom each annotation is attributed and when they recorded it.

CONCLUSIONS AND FUTURE WORK

Conclusions

In this paper we have described a unique system which was developed to enable the collaborative, real-time, indexing, browsing, annotation and discussion of high quality video content by multiple, distributed groups connected via access grid nodes on a broadband network.

Although previous video annotation systems have been developed, they have not been collaborative, real-time, synchronous systems capable of supporting high quality MPEG-2 content. These requirements have demanded that the collaborative application-sharing environment be developed from scratch using .NET Remoting. To ensure that the system is as flexible as possible, users are able to edit the Description Template directly. The user interface is then dynamically generated from the Template. For simplicity sake, our default metadata application profile is a simplified aggregation of particular MPEG-7 Description Schemes which can easily map to MPEG-7. Metadata input is controlled through a backend XML Schema as well as controlled vocabularies associated with specific terms. Fine-grained metadata generation is streamlined through the integration of Mediaware's automatic scene change detection algorithm. In order to maximize interoperability and leverage existing servers, we have chosen to extend the existing W3C Annotea tools for annotating web pages, to enable the annotation of audiovisual content.

There is enormous interest in this application – in particular from the medical and biological imaging domain for the annotation of bio-medical video content. Our goal is to use this tool to assist with the manual indexing by domain-experts of example databases which can then be used for machine learning to enable automatic domain-specific video recognition.

Future Work

In the next 12-18 months we intend to continue the development of the Vannotea system. In particular we would like to improve and extend it by implementing the following functionalities and carrying out the following tasks:

- Enable the attachment of audio/video annotations;
- Perform user evaluations and usability studies to obtain user feedback and refine and modify the software accordingly;
- Enable the sharing and annotation of documents of all media types (not just video) e.g. word documents, web pages, images, presentations, texts;
- Enable collaborative *editing* of documents of all media types;
- Investigate software and standards (MPEG-21) for managing the digital rights associated with the video content being delivered and annotated.

REFERENCES

[1] GrangeNet. http://www.grangenet.net/

[2] Access Grid. http://www.accessgrid.org/

[3] IBM MPEG-7 Annotation Tool. http://www.alphaworks.ibm.com/tech/videoannex

[4] Ricoh MovieTool. http://www.ricoh.co.jp/src/multimedia/MovieTool/

[5] Zentrum fuer Graphische Datenverarbeitung e.V. (ZGDV). VIDETO - Video Description Tool. http://www.rostock.zgdv.de/ZGDV/Abteilungen/zr2/Produkte/videto/index_html_en

[6] Swiss Federal Institute of Technology (EPFL). COALA (Content-Oriented Audiovisual Library Access) – LogCreator. http://coala.epfl.ch/demos/demosFrameset.html

[7] CSIRO. The Continuous Media Web (CMWeb). http://www.cmis.csiro.au/cmweb/

[8] David Bargeron, Anoop Gupta, Jonathan Grudin, and Elizabeth Sanocki. "Annotations for Streaming Video on the Web: System Design and Usage Studies". Microsoft Research, Redmond. http://www.research.microsoft.com/research/coet/MRAS/WWW8/paper.htm

[9] Annodex. http://annodex.nsw.cmis.csiro.au/index.html

[10] JJ Cadiz, Anand Balachandran, Elizabeth Sanocki, Anoop Gupta, Jonathan Grudin, and Gavin Jancke. "Distance Learning Through Distributed Collaborative Video Viewing", Technical Report, Microsoft Research, Redmond, May 10th, 2000. http://research.microsoft.com/research/coet/DTVI/TRs/paper.pdf

[11] NetMeeting. http://www.microsoft.com/windows/netmeeting/

[12] Asim Karim. "H.323 and Associated Protocols", November 26,1999. ftp://ftp.netlab.ohio-state.edu/pub/jain/courses/cis788-99/h323/index.html

[13] Screensound Australia. http://www.screensound.gov.au

[14] Australian Centre for Moving Image (ACMI). http://www.acmi.net.au

[15] Dublin Core. http://dublincore.org/

[16] MPEG-7. http://www.mpeg-industry.com/

[17] J.Hunter, "An Application Profile which combines Dublin Core and MPEG-7 Metadata Terms for Simple Video Description", July 2002. http://www.metadata.net/harmony/video_appln_profile.html

[18] W3C Annotea Project. http://www.w3.org/2001/Annotea/

[19] http://www.w3.org/2000/10/annotation-ns#

[20] W3C XML Pointer, XML Base and XML Linking. http://www.w3.org/XML/Linking

[21] Marja-Riitta Koivunen and Ralph R. Swick. "Metadata Based Annotation Infrastructure offers Flexibility and Extensibility for Collaborative Applications and Beyond", W3C, MIT Laboratory for Computer Science, 26 September 2001. http://www.w3.org/2001/Annotea/Papers/KCAP01/annotea.html

[22] Zope Annotation Server. http://www.zope.org/Members/Crouton/ZAnnot/

[23] W3C - Perllib Annotations server. http://www.w3.org/1999/02/26-modules/User/Annotations-HOWTO

[24] Algae HOWTO. http://www.w3.org/1999/02/26-modules/User/Algae-HOWTO.html

[25] Virtual Network Computing (VNC). http://www.uk.research.att.com/vnc/

[26] Simon Robinson, Burt Harvey, Christian Nagel, Ollie Cornes, Karli Watson, Morgan Skinner, Jay Glynn, Zach Greenvoss and Scott Allen, Professional C# (2nd Edition), Wrox Press, 2002. pp.981 – 1025.

[27] Mediaware Solutions. http://www.mediaware.com.au/

ACKNOWLEDGEMENTS

The work described in this paper has been funded by the by the Cooperative Research Centre for Enterprise Distributed Systems Technology (DSTC) through the Australian Federal Government's CRC Programme (Department of Industry, Science and Resources), as well as the GrangeNet program funded by the Australian Federal Government's BITS Advanced Network Initiative (Department of Communications, Information Technology and the Arts).

APPENDIX A

A Screen Shot of the FilmEd Application within an Access Grid session.

APPENDIX A

A Screen Shot of the FilmEd Application within an Access Grid session.