Building and Indexing a Distributed Multimedia Presentation Archive using SMIL

Jane Hunter, jane@dstc.edu.au
Suzanne Little, s.little@dstc.edu.au

DSTC Pty Ltd, Uni. of Queensland, Qld, 4072, Australia. Phone +617 33654310, Fax +617 33654311

Abstract

This paper proposes an approach to the problem of generating metadata for composite mixed-media digital objects by appropriately combining and exploiting existing knowledge or metadata associated with the individual atomic components which comprise the composite object. Using a distributed collection of multimedia learning objects, we test this proposal by investigating mechanisms for capturing, indexing, searching and delivering digital online presentations using SMIL (Synchronized Multimedia Integration Language). A set of tools have been developed to automate and streamline the construction and fine-grained indexing of a distributed library of digital multimedia presentation objects by applying SMIL to lecture content from both the University of Qld and Cornell University. Using temporal information which is captured automatically at the time of lecture delivery, the system can automatically synchronize the video of a lecture with the corresponding Powerpoint slides to generate a finely-indexed presentation at minimum cost and effort. This approach enables users to search and retrieve relevant streaming video segments of the lecture based on keyword or free text searches within the slide content. The underlying metadata schema, the metadata processing/generation tools, distributed archive, backend database and the search, browse and playback interfaces which comprise the system are also described in this paper. We believe that the relatively low cost and high speed of development of this apparently sophisticated multimedia archive with rich search capabilities, provides evidence to support the validity of our initial proposal.

Keywords: SMIL, multimedia, metadata, flexible delivery, online presentations

Table of Contents

1. Introduction

The future will lead to many more compound multimedia documents on the web which combine text, image, audio and video in rich complex structured documents in which temporal, spatial, structural and semantic relationships exist between the components. The problems associated with indexing, archiving, searching, browsing and retrieving these kinds of structured dynamic documents initially appear to be infinitely more complex than the resource discovery of simple atomic textual documents. However we propose that by exploiting the complementary and rich nature of the existing knowledge or metadata provided by each of the separate atomic components, it may actually be relatively easy to generate high quality fine-grained metadata for the composite mixed-media objects. Figure 1 demonstrates the theory underlying the work described in this paper - that high quality metadata for a composite object can be derived by exploiting the existing metadata associated with the individual components together wth the knowledge of the spatio-temporal relationships between them and that the combined effect of the metadata provided by each of the atomic objects may be greater than the sum of their parts.

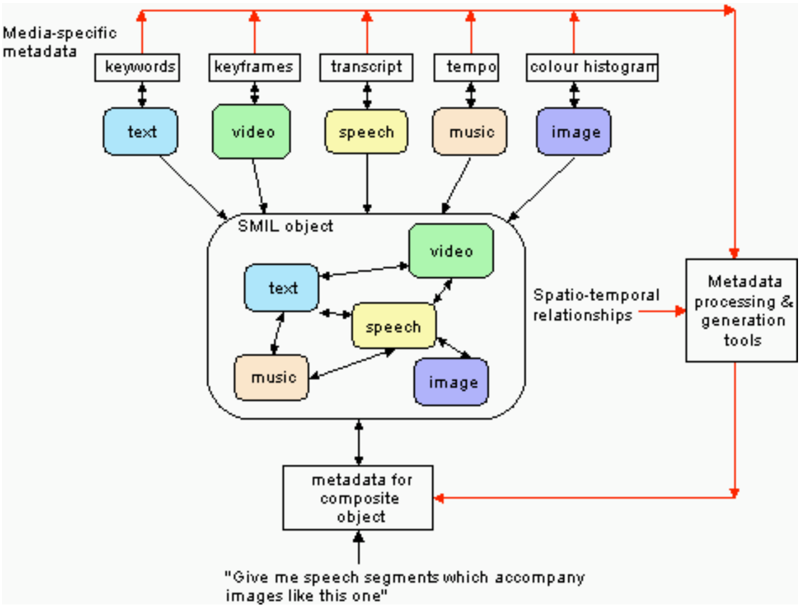


Figure 1 - Generation of Metadata for Composite Mixed-Media Digital Objects

To test this proposal, we developed a test bed of multimedia learning objects which use the W3C Recommendation SMIL 1.0 (Synchronised Multimedia Integration Language) [1] to combine video, powerpoint images and text from actual university lectures from the University of Queensland and Cornell University. This paper describes the complete system - from the filming of lectures, to the automatic SMIL object generation and archival, the metadata processing and generation tools and the search, browse and retrieval interface.

In addition to being a testbed for our multimedia metadata research, this distributed collection could be futher developed into a valuable distance learning resource. Universities and other educational organisations, are finding that both students and lecturers are demanding more flexible delivery mechanisms which allow them to peruse and deliver digital multimedia educational resources at any time and from any place. There are also growing markets for course material which can be packaged and sold to overseas universities or other educational organisations. Consequently there is huge potential commercial benefit for those universities and organisations who are able to cost-effectively provide such flexible learning/teaching environments through the internet.

However a fundamental requirement of distance learning systems is that they must be as platform and network independent as possible to allow access and dissemination to as wide an audience as possible. Access to high quality information and knowledge should not be dependent on expensive high bandwidth networks or proprietary software or hardware, which is beyond the price range of students.

For an archive of presentations to be an effective learning tool, it also requires fast, efficient, precise search facilities at various levels - across distributed collections as well as within individual presentations. In order to provide these services, standardized and well-modelled multi-level metadata is required. Metadata descriptions are required at the collection, presentation and slide/segment level. Segment-level metadata enables students to search on particular keywords and to find the individual slide or corresponding segment of the lecture video in which this topic was discussed. The high cost of manual transcription and the limitations of automatic speech recognition systems, make indexing of lectures via transcript unviable. However the textual content of the PowerPoint slides provides almost as powerful a resource as the transcript. In this project, we provide mechanisms for automating the temporal alignment of the PowerPoint slide content with the video of the lecture. We also provide metadata input tools to streamline the additional manual entry of descriptive or administrative metadata such as the lecturer, department, subject, date/time, place etc.

A web-based search interface has been built which enables users to search and browse the archive both across the distributed collections as well as within individual presentations and to retrieve and play the matching streaming segment of the relevant SMIL file.

Figure 2 illustrates the metadata processing and generation components for this application.

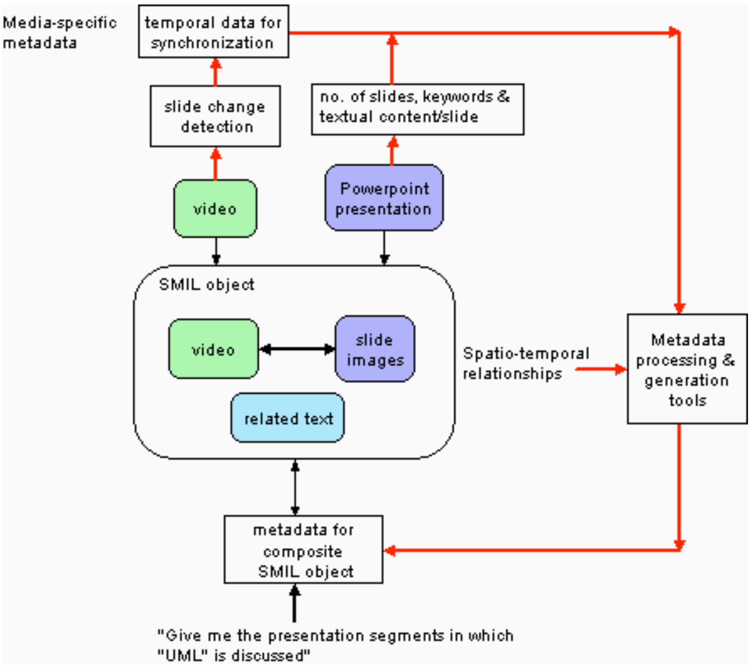


Figure 2 - Metadata Processing Steps for Lecture Archive

The remainder of the paper is as follows. In Section 2 we describe related work and the particular objectives and features of our system which differentiate it from other systems. Section 3 provides an overview of SMIL and a simple example to demonstrate the basic concepts. Section 4 describes the individual components of the system architecture, including the presentation capture process, the indexing and archival tools and the search, browse and retrieval interface. The paper concludes in Section 5 with an assessment of our ability to meet our objectives, an evaluation of our original proposal in the context of this digital multimedia collection and anticipated future work directions.

---

2. Evaluation of Related Work and Objectives

Early approaches to online learning involving digital video and audio primarily used CD-ROM or the internet to provide video or audio clips as a supplementary resource to written text. Typically the multimedia content was not integrated or synchronised with the textual resources and the indexing and search capabilities were very coarse.

A number of commercial products have recently appeared which enable the online publishing of lecture videos synchronised with PowerPoint presentations. These include Presenter.com [2], EZPresenter [3], Real Presenter Plus [4] and StreamAuthor [5]. These systems are typically expensive and depend on proprietary platform-dependent or application-dependent software to perform the synchronization or the replay. Presenter.com and EZPresenter lack the ability to automatically capture the synchronisation details and (apart from high-level browse interfaces) their indexing and search capabilities are negligible. They are also expensive which makes them inaccessible to many university departments. StreamAuthor does enable the video of the presentation to be synchronized with the PowerPoint presentation at the time of capture but does not support any search facilities other than high-level browsing across a single presentation. Real Presenter does provide an integrated table of contents for each presentation so viewers can jump ahead to a particular slide but it doesn't provide keyword or text searches across multiple presentations.

None of the systems utilize metadata standards to enable search and retrieval via a web search interface, across institutions, collections or within presentations.

Hence the objectives of the work described in this paper, which differentiate it from other similar systems, were as follows:

- To investigate mechanisms for automatically generating metadata for composite multimedia objects by utilizing existing metadata or knowledge associated with or extracted from the individual atomic digital objects;
- To build low-cost tools to streamline and automate the internet publishing of synchronised lectures and PowerPoint slides with links to other related course resources;
- To evaluate SMIL as a tool for marking-up and coordinating synchronised presentations of multimedia components;
- To build an online distributed collection of SMIL presentations using lecture content from the Uni. Of Qld and Cornell University, which is platform and network independent;
- To provide a search, browse and retrieval interface across the distributed archive at the collection, presentation and slide level, which supports keyword and text searches;
- To investigate the application of existing relevant metadata standards such as Dublin Core [6] (for resource discovery), IMS [7] (for educational resources) and MPEG-7 [8] (for multimedia content description), to satisfy the search and indexing requirements of this collection.

---

3. Overview of SMIL

Synchronized Multimedia Integration Language (SMIL 1.0) is a W3C Recommendation designed for choreographing web-based multimedia presentations which combine audio, video, text and graphics in real-time. It uses a simple XML-based markup language, similar to HTML, which enables an author to describe the temporal behavior of a multimedia presentation, associate hyperlinks with media objects and describe the layout of the presentation on a screen. The W3C SYMM (Synchronised Multimedia) Working Group recently released the SMIL 2.0 Working Draft [9] which extends the functionality contained in SMIL 1.0 by enabling interactivity in multimedia presentations, advancing the timing model and improving the accessibility features. In the work described here we have used SMIL 1.0 but when SMIL 2.0 moves to Recommendation stage and more SMIL 2.0 players become available then we plan to upgrade the system to SMIL 2.0.

SMIL has the following advantages to offer in the context of the application described in this paper:

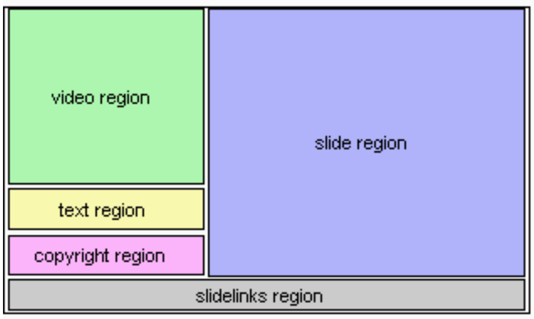
- Human-readability and similarity to HTML make SMIL easy to understand and use. Authoring can be done very cheaply and easily using simple text editors;
- Platform independence - as a W3C recommendation, SMIL is not a proprietary technology and hence does not tie the implementation to particular platforms or programming languages;
- Network and client adaptability - SMIL provides a *switch* tag which can be used to dynamically choose the most appropriate media object to stream, depending on client display capabilities or connection speed e.g., use audio instead of video for low bandwidths;
- Ready availability of SMIL players - there are nine SMIL 1.0 players available covering a wide range of platforms [10]. The most popular of these are Apple's QuickTime 4.1, Microsoft's Internet Explorer 5.5 Preview Browser and RealNetworks' RealPlayer 8.

3.1 Basic Concepts and a Simple Example

Generating a SMIL presentation consists of three basic steps: defining the regions for your media; linking media objects to those regions and determining the order, in which to play them (in sequence, parallel or some combination of both). The code below illustrates the skeleton code for a simple example SMIL file.

```
 <smil>
<head>
 <meta name="publisher" content="DSTC Pty Ltd"/>
 <meta name="date" content="2001-03-21"/>
 <layout>
  <!-- layout tags -->
 </layout>
</head>
<body>
 <!-- media and synchronization tags -->
</body>
</smil>
```

The *layout*, *root-layout* and *region* tags can be used to define spatial regions within a presentation (e.g., a video region, a slide region, a text region...) by specifying locations in pixels, relative to the top left hand corner or some other specified position. Figure 3 illustrates the SMIL code and the corresponding layout for a simple example which is very similar to our SMIL presentation replayer, shown in Figure 4 and described in Section 4.3.




```
<smil>
 <head>
  <layout>
    <root-layout width="851" height="395"/>
    <region id="videoregion" top="5" width = "350" height="262"/>
    <region id="slideregion" top="5" left="352" width = "499" height="355"/>
    <region id="textregion" top="267" width = "350" height="45"/>
    <region id="copyrightregion" top="313" width = "350" height="45"/>
    <region id="slidelinksregion" top="360" width = "851" height="35"/>
  </layout>
 </head>
 <body>
  <par>
   <img src="http://dstc.edu.au/Tut1/slide1.jpg" region="slideregion"/>
   <video src="rtsp://dstc.edu.au/Tut1/slide1.rm" region="videoregion"/>
   <textstream src=http://dstc.edu.au/Tut1.rt region="textregion"/>
   <textstream src="http://dstc.edu.au/copyright.rt" region="copyrightregion"/>
   <textstream src="http://dstc.edu.au/slideLinks.rt" region="slidelinksregion"/>
  </par>
 </body>
</smil>
```

Figure 3 - Spatial Layout for a Simple SMIL Example

The media and synchronization information is defined inside the *body* tag. Coarse-grained temporal structuring can be specified using the *seq* and *par* tags. The *seq* tag defines a sequence - its children are executed one after the other. The *par* tag specifies that its children must be executed in parallel. Fine grained synchronization is specified using the *dur* and *begin* tags. In the *body* section of our example in Figure 3, all of the media components are contained within a *par* tag because they are synchronized to replay in parallel, starting at the same time. The media type and a URL to the media object associated with each region ID are specified inside the temporal structuring tags.

Figure 4 below illustrates the replay interface for the SMIL presentations generated by our system and replayed using RealNetworks RealPlayer 8. It shows the first slide from a lecture by Dr Susan Hamilton from the Biological Sciences Faculty at the University of Qld.

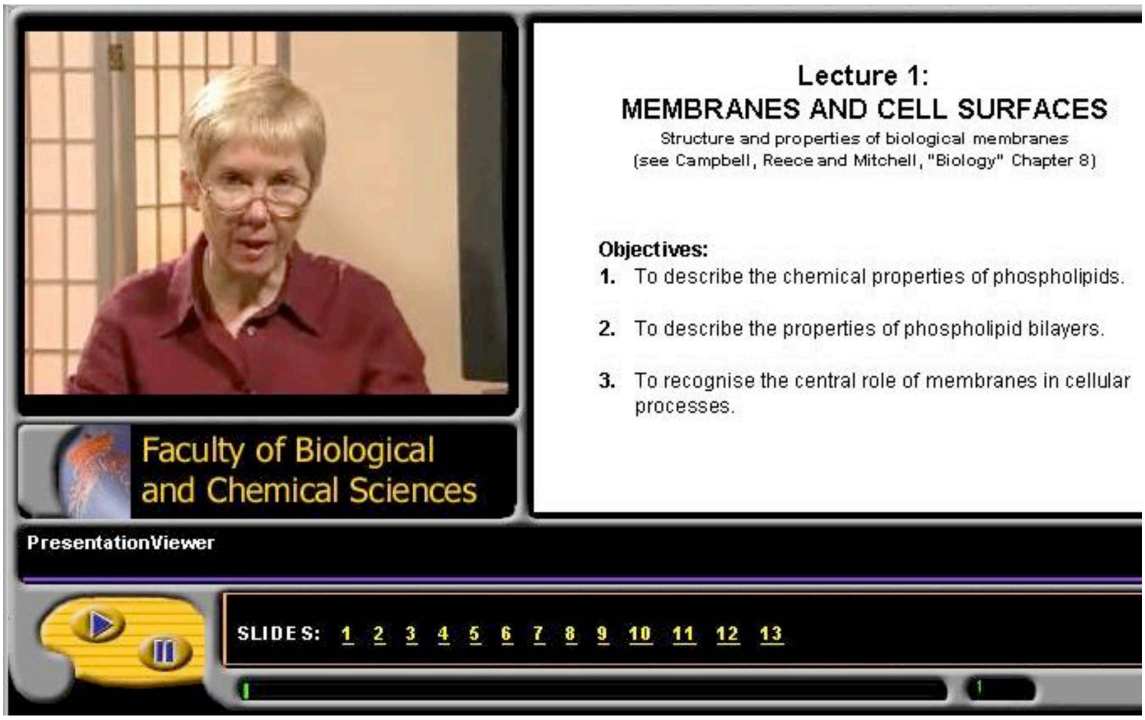


Figure 4 - Replay Interface for a SMIL Presentation Lecture

---

4. The System Components

This section describes the separate components of the system and how they fit into the overall process of building a searchable online presentation archive. Figure 5 provides an overview of the various components and the processes involved.

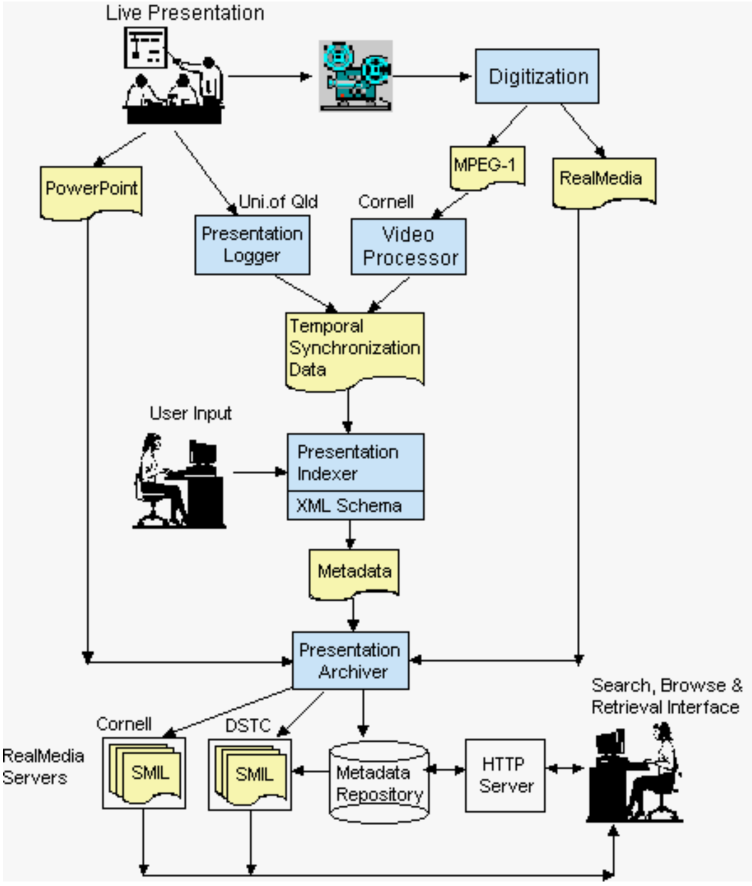


Figure 5 - Overview of System Components and Process Flow

4.1 Presentation Capture

The presentation capture process involves the following steps:

1. Filming a lecturer giving a PowerPoint presentation;
2. Digitizing and encoding the video footage to MPEG format (for analysis) and RealMedia format (for streaming);
3. Analysing the digitized footage or logging data to determine the times at which slide changes occur;
4. Using the temporal information to generate a SMIL presentation which integrates synchronized digital video and PowerPoint slides with a timeline for browsing and jumping directly to a particular slide.

The Cornell University and the University of Queensland/DSTC systems adopt different approaches to the filming and segmentation/synchronization processes.

Cornell uses two cameras set up in the lecture hall to capture the video footage. The *overview* camera captures the entire lecture dais from which the presenter lectures. The *tracking* camera, which contains a built-in hardware tracker which follows the speaker, captures a head-and-shoulders shot of the presenter. At the end of the lecture, the footage is transmitted over a network to a *processing server*. The lecturer also uploads the PowerPoint slides to this server. The processing server then uses image recognition techniques to match the PowerPoint slides to the sequences of video frames and to determine the times at which slide changes occur. Details of Cornell's LectureBrowser system are described in [11]. Our system takes the video footage, PowerPoint slides and timing information generated by the Cornell processing server and generates a log file consistent with the output from the *PresentationLogger*.

DSTC, on the other hand, uses a *PresentationLogger* (Windows NT) application which runs in the background during the presentation recording the time at which PowerPoint slide changes occur. The application logs the slide number, the time it appeared, the keywords of the slide (extracted from the title), and the full text of the slide (extracted from the text boxes on the slide). It also records the PowerPoint file location, the duration of the entire presentation and the date of the presentation (from the system time of the machine the presentation was given on).

The advantage of DSTC's approach is that the system can run in any lecture room, as long as the PresentationLogger software has been installed on the PC or laptop being used to give the presentation. The disadvantage is that only one manually-operated camera is used. Although manning the camera is labor-intensive, it also means better video quality since the camera settings can be adjusted to suit variations in lighting, rapid lecturer movement etc. This is supported by the conspicuous difference in video quality between Cornell University content (which uses unmanned cameras) and the University of Qld. lectures (which have been manually filmed).

4.2 Presentation Indexing and Archival

The *PresentationIndexer* is a Visual Basic application which provides a user interface for the manual entry of additional descriptive and administrative metadata which cannot be extracted or inferred from the component media objects. Metadata input includes the location of the log file generated by the PresentationLogger plus details such as the presentation location, lecturer, department, institution, the presentation description, and metadata for the collection that this presentation belongs to (if this hasn't already been entered). To speed up metadata entry, a behind-the-scenes database is using default values and previously entered data to populate fields ahead of the user.

A metadata model was developed which enables the resource discovery of three entity types, corresponding to three levels of granularity:

- Collections;
- Presentations;
- Segments.

The properties associated with each of the entities are based on the Dublin Core Element Set [12] but with additional properties to cover the educational and multimedia aspects of the content. An XML Schema [13] corresponding to the metadata model was developed and is listed in Appendix A.

Figure 6 below is a screen dump of some of the metadata input forms which comprise the *PresentationIndexer* application.

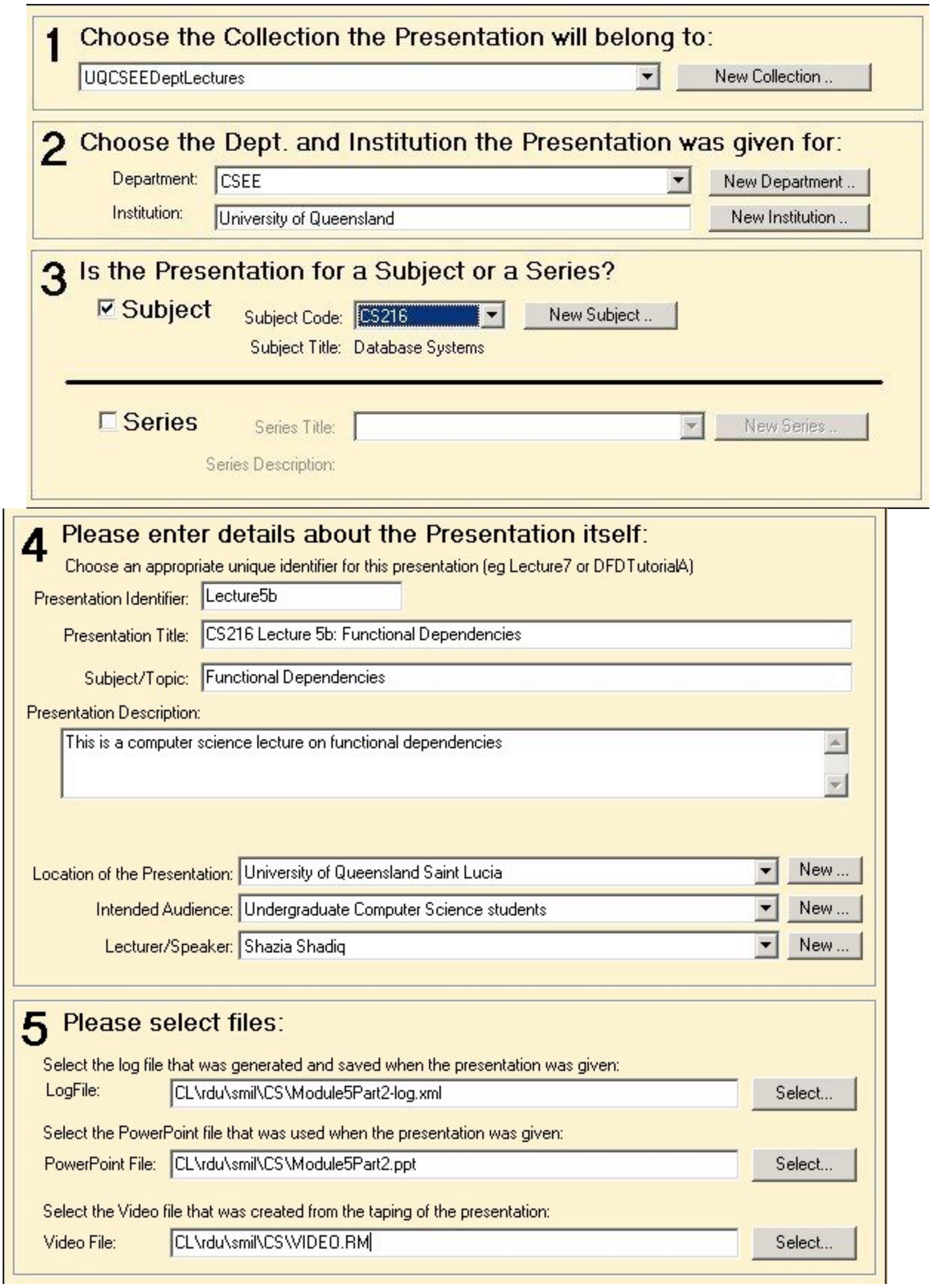


Figure 6 - Presentation Indexer Application

Before the description is saved to an XML file, the XML Spy processor [14] checks that the metadata input conforms to the XML Schema. If the metadata description does not conform, appropriate error messages are generated. An example of a metadata description which conforms to this schema is given in Appendix B.

The *PresentationArchiver* program takes the validated XML output from the *PresentationIndexer*, together with the associated PowerPoint and RealMedia files and generates a synchronised SMIL presentation. The SMIL files are saved to specified server locations at either DSTC or Cornell (or to CD-ROM for distribution). Alternatively web server scripts can regenerate the SMIL presentations dynamically from the metadata (which is stored in a database) and serve them across the web.

The associated metadata descriptions are saved to a database with links to the associated SMIL file and component objects. Alternatively the metadata could be embedded within the header of the SMIL file using the *meta* tag, as illustrated in the skeleton SMIL example in Section 3.1. SMIL 2.0 provides a *metadata* tag [15] which is capable of supporting more complex structured multimedia metadata descriptions (e.g., MPEG-7 descriptions) embedded within the SMIL file header.

4.3. The Search, Browse and Retrieval Interface

The Web search, browse and retrieval interface to the distributed presentation archive can be accessed at [16]. In order to replay retrieved presentations or presentation segments, users need to have the RealPlayer plugin [17] installed on their workstation. The Web interface enables users to browse the complete list of presentations [18], or to perform free text or keyword searches:

- over the complete distributed archive (both Cornell and DSTC),
- within individual collections (either Cornell or DSTC), or
- within particular presentations.

Figure 7 shows the search interface to the archive and Figure 8 shows the corresponding results page with the matching query string "***project***" highlighted. Users can choose to replay just the presentation segment which contains the query string or they can replay the complete presentation.

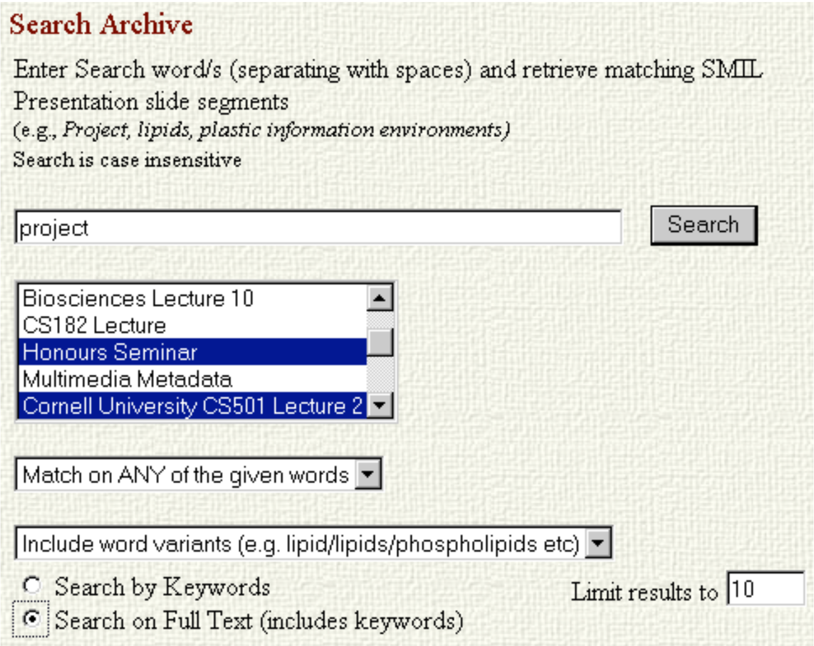


Figure 7 - Presentation Archive Search Interface

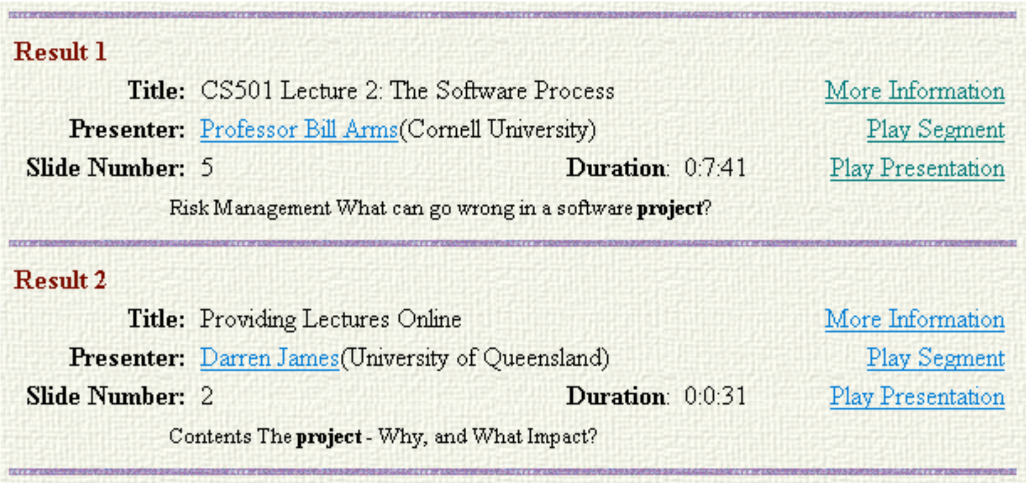


Figure 8 - Search Results

Server scripts dynamically generate the relevant SMIL segments from the stored metadata descriptions which reference the media content stored on the Real Media servers (the streaming video/audio components) and the web server (the text and slide images). Either the complete SMIL presentation or the selected segment is then streamed to the client and the RealPlayer plugin is invoked. Figure 9 illustrates the browse and replay interface for a retrieved presentation. Users can play the video (using the buttons in the left hand corner) and the corresponding slides will appear automatically or they can jump to any slide number and the corresponding video component will begin playing.

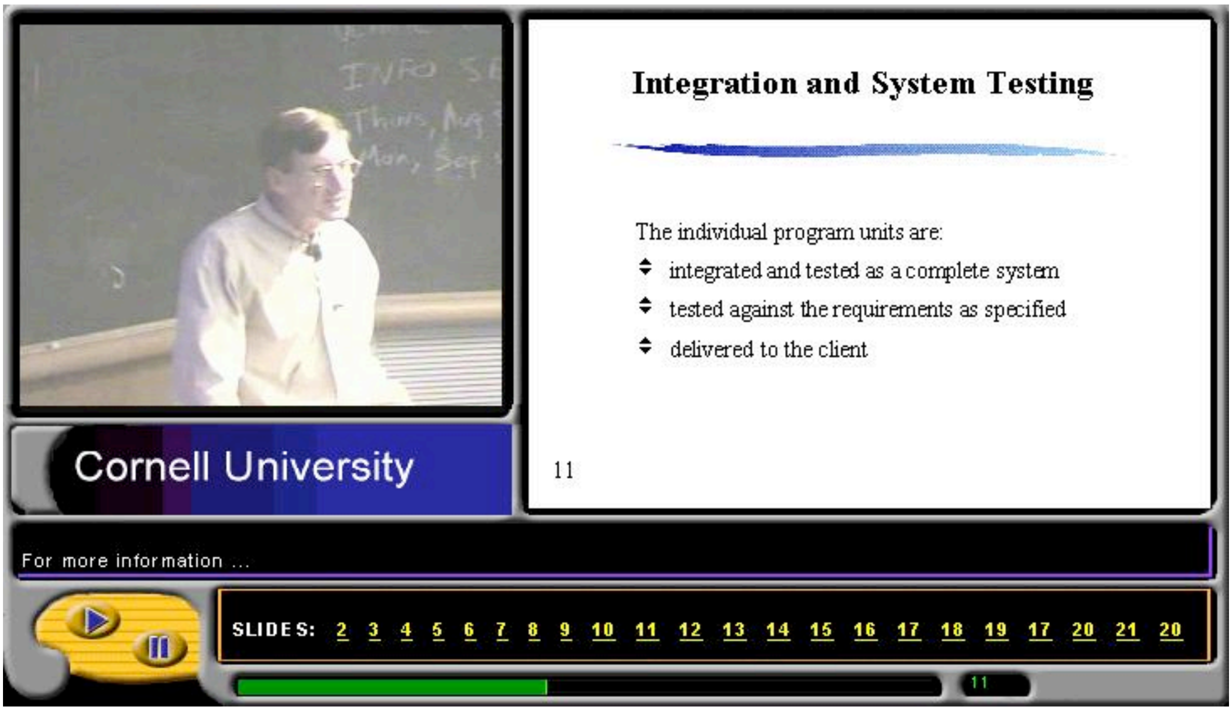


Figure 9 - Presentation Browse and Replay Interface

---

5. Conclusions and Future Work

5.1 Conclusions

In this paper we have presented an automated, cost-effective system for developing a distributed archive of indexed multimedia presentations. By appropriately combining complementary metadata derived from the individual digital objects, we have been able to provide fine-grained, free-text and keyword search and retrieval across multimedia presentations.

Metadata generation is often a prohibitively expensive exercise. During the development of this system, we have demonstrated how the costs of generating high-quality, fine-grained metadata for complex objects can be reduced by:

- generating metadata at the time of content creation. Both the DSTC and Cornell systems capture the temporal, as well as certain descriptive and administrative metadata at the time of lecture delivery and capture;
- applying automatic feature extraction techniques to generate metadata. The Cornell system applies image recognition techniques

to the digitized video to determine precisely when slide changes occur. This enables automation of the synchronization and temporal indexing processes;
- exploiting any pre-existing, default or duplicated metadata values where possible e.g., contextual metadata such as institution, lecturer, course etc.;
- inferring metadata where possible. For example, the size of the collection is automatically incremented each time a new lecture is added;
- constraining metadata input via an XML Schema (Appendix A). By validating and constraining metadata input to the structures and data types defined within the specified schema, the quality of the metadata descriptions can (to a certain extent) be controlled;
- constraining metadata input to controlled vocabularies. A number of pull-down lists are built into the PresentationIndexer application to constrain input to values from a set of controlled terms or a thesaurus;
- using standards (such as SMIL and Dublin Core) which maximize access, re-use and interoperability.

Our evaluations of SMIL 1.0 have shown that, although it provides a simple, platform independent and network independent approach to specifying and delivering composite mixed-media objects, it has certain limitations. These include limited text support, hyperlink support and interactivity capabilities within presentations. However we expect the extended functionality of SMIL 2.0 (which is expected to be published as a W3C Recommendation later this year) to overcome the majority of these limitations.

5.2 Future Work

Plans for future work include investigating the development of presentation editing tools which permit updates, additions and concatenations of existing presentations without loss of synchronisation. Functionality such as *cut*, *paste* and *delete* of presentation segments would enable educators to: reuse existing presentation segments, update presentations with more recent or topical material or customize presentations for particular audiences, without having to refilm the entire lecture each time.

Future work also includes upgrading the system to SMIL 2.0 when it progresses to a W3C Recommendation and when more players which support the extended functionality of SMIL 2.0, become available. This would allow the integration of a interactive search facilities and hyperlinks to related resources within the presentations, providing even greater flexibility to students, enabling them to navigate through presentations according to their particular needs.

Acknowledgements

The authors wish to acknowledge the use of lecture material belonging to the University of Queensland and Cornell University. We also wish to acknowledge the generosity and cooperation of the presenters whose lecture content is contained within the archive: Bill Arms, Simon Kaplan, Susan Hamilton, Shazia Shadiq and Darren James.

The authors also wish to acknowledge the valuable contributions which the Harmony project collaborators, Carl Lagoze and Dan Brickley have made to this work.

The authors also wish to acknowledge that this work was carried out within the Cooperative Research Centre for Research Data Networks established under the Australian Government's Cooperative Research Centre (CRC) Program and acknowledge the support of CITEC and the Distributed Systems Technology CRC under which the work described in this paper is administered. Travel funding for this work was provided by a grant approved under the Technology Diffusion Program of the Commonwealth of Australia's Department of Industry, Science and Resources.

References

[1] Synchronized Multimedia Integration Language (SMIL 1.0) Specification, W3C Recommendation, 15 June 1998, http://www.w3.org/TR/REC-smil/

[2] Presenter Inc., http://www.presenter.com/

[3] Vision 360 EZPresenter, http://www.vision360.net/companyinfo/01.html

[4] Real Presenter Plus, http://www.realnetworks.com/products/presenterplus/index.html

[5] StreamAuthor, http://ipw.internet.com/development/rich_media/985015125.html

[6] Dublin Core Metadata Initiative, http://www.dublincore.org/

[7] IMS Metadata Specification 1.1, June 2000, http://www.imsproject.org/metadata/

[8] MPEG-7 Home Page, http://www.darmstadt.gmd.de/mobile/MPEG7/index.html/

[9] Synchronized Multimedia Integration Language (SMIL 2.0) Specification , W3C Working Draft, 01 March 2001, http://www.w3.org/TR/smil20/

[10] SMIL Players , http://www.w3.org/AudioVideo/#SMIL

[11] Sugata Mukhopadhyay and Brian Smith, Passive Capture and Structuring of Lectures. , ACM Conference on Multimedia, 1999

[12] Dublin Core Element Set , http://www.dublincore.org/documents/dces/

[13] XML Schema Language, http://www.w3.org/XML/Schema

[14] XML Spy 3.5 , http://www.xmlspy.com/

[15] The SMIL 2.0 metadata element , http://www.w3.org/TR/smil20/metadata.html#edef-metadata

[16] SMIL Presentation Archive Search Interface , http://sunspot.dstc.edu.au:8888/smilsearch/search.html

[17] RealPlayer 8 , http://www.real.com/playerplus/index.html

[18] SMIL Presentation Archive Browse Interface , http://sunspot.dstc.edu.au:8888/smilsearch/search.html#browse

---

Appendix A: XML Schema for the Presentation Archive

```
<?xml version="1.0"?>

  <!--  This schema (version 3.2) represents the metadata associated with
  a collection of presentations that each consists of a synchronized video
  and powerpoint presentation. -->

   <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
        <xsd:annotation>
            <xsd:documentation>
                    Metadata Schema for Presentation Archive
            </xsd:documentation>
        </xsd:annotation>

        <xsd:element name="PresentationCollection" type="PresentationCollection"/>
        <xsd:complexType name="PresentationCollection">
            <xsd:sequence>
                    <xsd:element name="Title" type="xsd:string"/>
                    <xsd:element name="Creator" type="xsd:string"/>
                    <xsd:element name="Subject" type="xsd:string"/>
                    <xsd:element name="Description" type="xsd:string"/>
                    <xsd:element name="Publisher" type="xsd:string"/>
                    <xsd:element name="DateCreated" type="xsd:date"/>
                    <xsd:element name="Format" type="xsd:string"/>
                    <xsd:element name="Language" type="xsd:language"/>
                    <xsd:element name="Rights" type="xsd:string"/>
                    <xsd:element name="Size" type="xsd:integer"/>
                    <xsd:element name="Presentation" type="PresentationType"
                                 maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:attribute name="id" type="xsd:uriReference"/>
        </xsd:complexType>

        <xsd:complexType name="PresentationType">
            <xsd:sequence>
                    <xsd:element name="Title" type="xsd:string"/>
                    <xsd:element name="Subject" type="xsd:string"/>
                    <xsd:element name="Description" type="xsd:string"/>
                    <xsd:element name="Presenter" type="PresenterType"/>
                    <xsd:element name="Date" type="xsd:date"/>
                    <xsd:element name="Place" type="xsd:string"/>
                    <xsd:element name="PresentationNumber" type="xsd:integer"/>
                    <xsd:element name="Duration" type="xsd:string"/>
                    <xsd:choice>
                            <xsd:element name="Course" type="CourseType"/>
                            <xsd:element name="Series" type="SeriesType"/>
                    </xsd:choice>
                    <xsd:element name="Department" type="xsd:string"/>
                    <xsd:element name="Institution" type="xsd:string"/>
                    <xsd:element name="PresentationVideo"
                                 type="PresentationVideoType"/>
                    <xsd:element name="PresentationSlides"
                                 type="PresentationSlidesType"/>
            </xsd:sequence>
            <xsd:attribute name="id" type="xsd:uriReference"/>
        </xsd:complexType>
```

```
    <xsd:complexType name="PresenterType">
        <xsd:sequence>
                <xsd:element name="Name" type="xsd:string"/>
                <xsd:element name="Title" type="xsd:string"/>
                <xsd:element name="Affiliation" type="xsd:string"/>
                <xsd:element name="Email" type="xsd:string"/>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="CourseType">
        <xsd:sequence>
                <xsd:element name="SubjectCode" type="xsd:string"/>
                <xsd:element name="SubjectDescription" type="xsd:string"/>
                <xsd:element name="Semester" type="xsd:string"/>
                <xsd:element name="CourseCode" type="xsd:string"
                                 minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element name="CourseDescription" type="xsd:string"
                                 minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="SeriesType">
        <xsd:sequence>
                <xsd:element name="SeriesCode" type="xsd:string"/>
                <xsd:element name="SeriesDescription" type="xsd:string"/>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="PresentationVideoType">
        <xsd:sequence>
                <xsd:element name="Format" type="xsd:string"/>
                <xsd:element name="Duration" type="xsd:string"/>
                <xsd:element name="Rights" type="xsd:string"/>
                <xsd:element name="VideoSegment" type="VideoSegmentType"
                                 maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:uriReference"/>
    </xsd:complexType>

    <xsd:complexType name="PresentationSlidesType">
        <xsd:sequence>
                <xsd:element name="Format" type="xsd:string"/>
                <xsd:element name="Size" type="xsd:integer"/>
                <xsd:element name="Rights" type="xsd:string"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:uriReference"/>
    </xsd:complexType>

    <xsd:complexType name="VideoSegmentType">
        <xsd:sequence>
                <xsd:element name="Keyword" type="xsd:string"
                                 minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element name="Text" type="xsd:string" minOccurs="0"/>
                <xsd:element name="StartTime" type="xsd:string"/>
                <xsd:element name="EndTime" type="xsd:string"/>
                <xsd:element name="Duration" type="xsd:string"/>
                <xsd:element name="SlideImage" type="xsd:uriReference"
                                 minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:string"/>
    </xsd:complexType>

</xsd:schema>
```

Appendix B: An Example XML Description for a Presentation

```
<PresentationCollection id="CornellCSDeptLecturesFall2000"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="http://dstc.edu.au/smil/schema.xml">
        <Title>Cornell University CS Dept. Lectures</Title>
        <Creator>Cornell University CS Dept.</Creator>
        <Subject>CS Dept. Lectures</Subject>
        <Description>
          Series of lectures from the Cornell University CS Dept.
        </Description>
        <Publisher>Cornell University CS Dept.</Publisher>
        <DateCreated>2001-02-26</DateCreated>
        <Format>application/smil</Format>
        <Language>en</Language>
        <Rights>Cornell University</Rights>
```

```xml
        <Size>7</Size>
        <Presentation id="http://cs.cornell.edu/lecture_archive/Lecture7/LECTURE7.SMI">
            <Title>CS501 Lecture 7: Management II: Business and Legal Aspects of Software Engineering</Title>
            <Subject>Management II: Business and Legal Aspects of Software Engineering</Subject>
            <Description>
               This lecture covers the business and legal aspects of
               management in software engineering
            </Description>
            <Presenter>
                <Name>Bill Arms</Name>
                <Title>Professor</Title>
                <Affiliation>Cornell University</Affiliation>
                <Email>wya@cs.cornell.edu</Email>
            </Presenter>
            <Date>2000-09-14</Date><!-- T12:24 -->
            <Place>Phillips Hall 101, Cornell University</Place>
            <PresentationNumber>7</PresentationNumber>
            <Duration>P0Y0M0DT1H15M3S</Duration>
            <Course>
                <SubjectCode>CS501</SubjectCode>
                <SubjectDescription>Software Engineering</SubjectDescription>
                <Semester>Fall 2000</Semester>
                <CourseCode>B.Comp.Sci.</CourseCode>
                <CourseDescription>Undergraduate Computer Science</CourseDescription>
            </Course>
            <Department>CS Dept</Department>
            <Institution>Cornell University</Institution>
            <PresentationVideo id="rtsp://cs.cornell.edu/cs501/Lecture7/cs501.rm">
                <Format>RealVideo</Format>
                <Duration>P0Y0M0DT1H15M3S</Duration>
                <Rights>Cornell University</Rights>
                <VideoSegment id="1">
                        <Keyword>CS</Keyword>
                        <Keyword>501:</Keyword>
                        <Keyword>Software</Keyword>
                        <Keyword>Engineering</Keyword>
                        <Keyword>Fall</Keyword>
                        <Keyword>2000</Keyword>
                        <Text>CS 501: Software Engineering
                              Fall 2000 Lecture 7
                              Management II
                              Business and Legal Aspects of
                              Software Engineering</Text>
                        <StartTime>0:0:14</StartTime>
                        <EndTime>0:2:0</EndTime>
                        <Duration>0:2:46</Duration>
                        <SlideImage>http://cs.cornell.edu/lecture_archive/Lecture7/Slide1.JPG</SlideImage>
                </VideoSegment>
                <VideoSegment id="2">
                        <Keyword>Administration</Keyword>
                        <Text>Administration
                              Read instructions for Assignment 2
                              Schedule presentation
                              Group and individual assignment
                              No office hours next Tuesday
                              No recitation session on Monday
                              See Notices for Teaching Assistant
                              assignments to projects.</Text>
                        <StartTime>0:2:0</StartTime>
                        <EndTime>0:6:11</EndTime>
                        <Duration>0:4:11</Duration>
                        <SlideImage>http://cs.cornell.edu/lecture_archive/Lecture7/Slide2.JPG</SlideImage>
                </VideoSegment>
                <VideoSegment id="3">
                        <Keyword>Legal</Keyword>
                        <Keyword>Environment</Keyword>
                        <Text>Legal Environment Software is
                              developed in a complex legal and economic
                              framework. Changes in laws follow changes
                              in technical world.
                              Jurisdictions:
                                United States Constitution
                                International treaties
                                Federal and state statues
                                Precedents
                                Supreme Court
                                Cost of establishing precedent</Text>
                        <StartTime>0:6:11</StartTime>
                        <EndTime>0:10:25</EndTime>
                        <Duration>0:4:14</Duration>
                        <SlideImage>http://cs.cornell.edu/lecture_archive/Lecture7/Slide3.JPG</SlideImage>
                </VideoSegment>
```

```
                <VideoSegment id="4">
                        <Keyword>Legal</Keyword>
                        <Keyword>Topics</Keyword>
                        <Text>Legal Topics
                              International
                              Intellectual property (copyright, patent, contract)
                              Tort (e.g., liability of Internet service provider)
                              Privacy
                              Free speech and its limitations (government secrets, obscenity, blasphemy, hate)
                              Legal Information Institute: http://www.law.cornell.edu/</Text>
                        <StartTime>0:10:25</StartTime>
                        <EndTime>0:12:27</EndTime>
                        <Duration>0:2:2</Duration>
                        <SlideImage>http://cs.cornell.edu/lecture_archive/Lecture7/Slide4.JPG</SlideImage>
                </VideoSegment>

......
......

                <VideoSegment id="18">
                        <Keyword>Fixed</Keyword>
                        <Keyword>Variable</Keyword>
                        <Keyword>Costs:</Keyword>
                        <Keyword>Profit</Keyword>
                        <Keyword>Loss</Keyword>
                        <Text>Fixed and Variable Costs:
                             Profit or Loss $15M $10M $5M Unit sales
                             2,500 5,000 7,500</Text>
                        <StartTime>1:10:58</StartTime>
                        <EndTime>1:15:03</EndTime>
                        <Duration>0:5:5</Duration>
                        <SlideImage>http://cs.cornell.edu/lecture_archive/Lecture7/Slide18.JPG</SlideImage>
                </VideoSegment>
            </PresentationVideo>

            <PresentationSlides id="http://cs.cornell.edu/lecture_archive/Lecture7/Lecture7.ppt">
                <Format>Microsoft PowerPoint</Format>
                <Size>18</Size>
                <Rights>Professor Bill Arms</Rights>
            </PresentationSlides>

        </Presentation>

</PresentationCollection>
```