

# A Streamlined System for Building Online Presentation Archives using SMIL

**Darren James**  
**DSTC Pty Ltd**  
**University of Queensland**  
**QLD 4072, Australia**  
**darren.james@dstc.edu.au**

**Jane Hunter**  
**CITEC**  
**Brisbane**  
**QLD 4001, Australia**  
**jane@dstc.edu.au**

## **Abstract**

The demand for and expectation of ubiquitous access to multimedia online learning resources are much higher amongst computer science students than other fields of study. Previous systems providing internet access to digital video or audio recordings of lectures have been disappointing or ineffective as a learning experience or excessively complex and time-consuming from the educator's point of view. This paper describes a new approach to building an online presentation archive of lectures, based on a relatively new Recommendation from W3C called SMIL (Synchronized Multimedia Integration Language) [1]. It describes a highly efficient and integrated system for capturing, indexing, searching and delivering online presentations without relying upon proprietary technology.

**Keywords:** SMIL, metadata, video, flexible delivery, online presentations

## **1 Introduction**

Computer science students in particular are now demanding more flexible delivery mechanisms which allow them to peruse digital multimedia educational resources at any time and from any place. As well as ubiquitous access, the students expect advanced search interfaces which provide fast, efficient, fine-grained and precise search and retrieval of relevant online educational resources. The provision of such services requires detailed indexing which is often prohibitively expensive.

Previous approaches involving digital video and audio have primarily used CD-ROM or the internet to provide video or audio clips as a supplementary resource to written text. Usually the multimedia content is not integrated or synchronised with the textual resources and the indexing and search capabilities are very coarse.

Systems which do enable the integration of multiple multimedia resources (text, image, audio, video, animation, graphics) into synchronised presentations and also provide indexing and searching services over these presentations, are expensive and/or depend on proprietary platform- (or even application-) dependent software.

In this paper we describe a platform-independent, efficient and economically-viable approach to building a searchable

online multimedia presentation archive based on the SMIL and XML Schema [2] recommendations from W3C [3] and the Dublin Core metadata standard from DCMI [4].

The remainder of this paper is as follows. Section 2 provides an overview of similar commercial and research systems for flexible delivery of educational material. Section 3 describes the requirements for a successful system and the consequent objectives of this project. Section 4 describes the architecture and components of this system. Section 5 describes the use of SMIL for synchronising archived presentation elements. Section 6 describes feedback to the system as a result of its application within the University of Qld CSEE department, and in Section 7 we discuss future work plans.

## **2 Related Work**

Commercial products which enable the online publishing of lecture videos synchronised with PowerPoint presentations include Presenter.com [5] and Webpresenter [6]. Although both products have very slick interfaces, they lack the ability to automatically capture the synchronisation details and apart from a high-level browse interface, the indexing and search capabilities are negligible. They are also expensive which makes them inaccessible to many university departments.

A number of non-commercial systems have incorporated mechanisms to automate the synchronisation process. The Lecture Browser project at Cornell [7] uses multiple cameras in the lecture hall to track both the lecturer and the current position within the PowerPoint presentation. Image recognition techniques are then applied during post-processing to match the recorded video of the live PowerPoint presentation to the slides in the static PowerPoint file to determine synchronisation timestamps. This prototype does not consider standardized metadata requirements to enable retrieval via a web search engine, across collections or via fine-grained text or keyword searching within presentations.

The Arts Multimedia Centre at the University of Western Australia [8] aims to automate the synchronisation process by capturing the movements of the lecturers hands over an overhead transparency (using fixed camera unit) along with

the sound of the lecturer. No search facilities are provided within the lecture content.

The University of Texas Arlington, Center for Distance Education [9], provides internet access to digitised videos of lectures synchronised with PowerPoint slides – but again the search interface is limited to browsing a list of subjects.

The other limitation of most of these systems is that they focus on the head of the speaker and do not accommodate material other than PowerPoint presentations. We believe that the main benefit of having video in an electronic learning environment is not due to the "talking head" phenomena but comes from the ability to capture other visual information such as the physical interaction between the lecturer and resources in the lecture room e.g pointing to the projected slide points, writing on a white board or even demonstrations using physical models or experiments performed within the lecture room.

### **3 Project Objectives**

#### **3.1 Flexible Delivery**

Universities, along with many other educational institutions, are finding that their current (and potential) students are demanding more flexibility when it comes to attending tutorials and lectures. There are those who wish to further their education whilst still participating in full-time work that does not permit them to attend lectures. There are regional, interstate and overseas students who are not able to physically locate to the university. And there are the markets that universities are only beginning to exploit in which course material is packaged and sold to overseas universities, or other institutions. Hence there is a need (and potentially huge commercial benefit) within Universities to provide flexible learning environments based on the internet.

#### **3.2 Multimedia Content – more than just videos of lectures**

Electronic educational resources may include text, image, graphics, audio, video, animation and web resources. Creating an educational multimedia archive is not simply filming material being presented and then allowing students to view/listen to those videos/audio recordings. This has been done before and low student usage within libraries has demonstrated the ineffectiveness of this approach [10].

We believe that integrated multimedia presentations which combine text, image, graphics, audio, video, animation and web resources in meaningful synchronised ways will enhance the effectiveness of these resources as learning tools.

This project does not aim to replace live lectures in an educational context. While providing an efficient and simple tool for archiving and presenting presentations such as lectures and corporate seminars, it is meant to be used in conjunction with other educational resources to enable educators to easily supplement lecture material. This use includes producing multimedia tutorials of material that

students find difficult to grasp, so that they can review the material at a later time.

#### **3.3 Egalitarian Access**

A fundamental requirement of distance learning systems is that they must be as platform 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 expensive proprietary software or hardware, which is beyond the price range of students.

#### **3.4 Indexing and Search Requirements**

For an archive of presentations to be an effective learning tool, it needs to provide search facilities across its content as well as within individual presentations. If public access is desirable then the collection should also be indexed in such a way that generic web search engines can search across collections to retrieve presentations from within institutional collections.

In order to provide these services, standardized and well-modelled multi-level metadata is required. Metadata descriptions are required at the collection, presentation and segment level. By segments, we mean fine-grained components of presentations. Students require mechanisms which enable them to search on specific topics and find the individual slide or portion of the lecture in which this topic was discussed.

Clearly the transcript associated with a lecture would be the ideal indexing resource. But given the cost of manual transcription and the limitations of automatic speech recognition systems, this approach remains unviable for most academic departments.

However for fine-grained searching of a presentation archive's contents, the textual content of the PowerPoint slides provides almost as powerful a resource as the transcript. This text is already available in a digital form but is not linked temporally to the corresponding video or audio of the lecture. In this project, we provide a tool for automating the temporal alignment of the PowerPoint slide content and the lecture video.

We also provide metadata input tools to enable the manual entry of standardized high-level collection and presentation metadata descriptions which contain information about the lecturer, subject, place etc.

Indexing based on such metadata, will enable fast, efficient, fine-grained and precise search and retrieval both across and within the presentations [16].

#### **3.5 Summary of Goals**

The key research goals of the project are as follows:

- To build tools to streamline and automate the internet publishing of synchronised lectures and PowerPoint slides with any other relevant multimedia resources;

- To evaluate SMIL as a tool for marking-up and coordinating synchronised presentations of multimedia components;
- To build an archive test bed of Uni. Of Qld CSEE lectures, that offers platform and application independence where possible;
- To provide tools for the fast and efficient entry and generation of standardized Dublin Core-based metadata descriptions;
- To provide search and retrieval interfaces across presentation collections, within collections and within individual presentations.

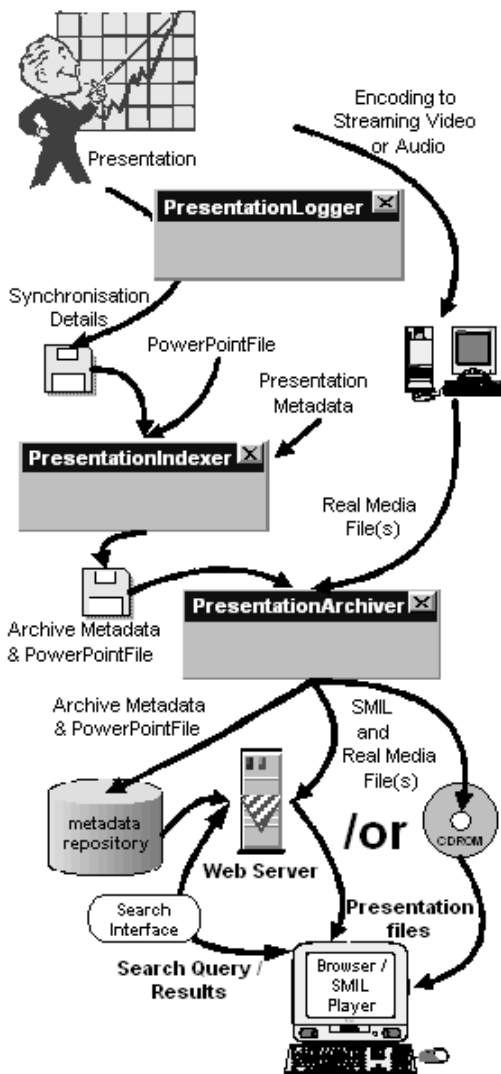


Figure 1 – Process Flow Using System Components

## 4 The System Architecture and Components

This section describes the components of the system and how they fit into the overall process of building a searchable online presentation archive. Figure 1 provides an overview of the various components and the steps involved.

### 4.1 PresentationLogger

The PresentationLogger is the application which logs the presentation in real time and records the time of the transitions between PowerPoint slides. It 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 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).

By automatically logging a presentation while it is being given (whether it is in the studio, the tutorial room or anywhere else), the time stamping and extraction of keywords and text reduces the indexing effort on the part of the archivers (or educators themselves) to effectively nil. It is this information that allows useful fine-grained searches such as full-text & keyword, to bring back results that can be pinpointed down to a slide within a presentation. So the student doesn't have to fast forward in search of the relevant material. They click on the search result, the presentation opens up for them, and it will automatically go straight to that portion of the presentation that is relevant.

**4 Please enter details about the Presentation itself:**  
 Choose an appropriate unique identifier for this lecture (eg Lecture7 or DFDTutorialA)

Presentation Identifier:

Presentation Title:

Presentation Description:

Location of the Presentation:

Intended Audience:

Lecturer/Speaker:

Enter some keywords that describe the lecture's content.  
 Keywords:

Figure 2 – A Screen Shot of Metadata Entry using the PresentationIndexer.

### 4.2 PresentationIndexer

The PresentationIndexer application provides a user interface for the manual entry of metadata associated with a collection or individual presentation. Input includes the location of the log file output by the PresentationLogger, the PowerPoint file from the presentation and details about the presentation location, the lecturer, the department and institution, the topic of the presentation, and the metadata for the collection that it belongs to. To speed up metadata entry, a behind-the-scenes database is populating fields ahead of the user based on previous entries and information entered so far.

The metadata fields are based on Dublin Core but with extensions for educational and multimedia content. An XML Schema is used to check that the metadata input conforms to the schema before it is saved. Appendix A shows the schema which we are using for this project. An example of a typical metadata description which conforms to this schema is given in Appendix B.

### 4.3 PresentationArchiver

The PresentationArchiver takes the XML output from the PresentationIndexer, and the encoded Real Media file (generated from the the video/audio taping) and creates synchronised presentations in the form of the SMIL files. The metadata descriptions are saved to a database, the SMIL files to a specified server location (or CD-ROM for distribution) and the Real Media files to a Real Media Server. Alternatively web server scripts can regenerate the SMIL presentations dynamically from the database tables to serve them across the web.

### 4.4 The LecturePresenter

The storage to a database (currently any Relational Database such as Oracle, but we are investigating XML Databases such as Lore [11]) enables the search and retrieval of metadata by server scripts invoked by a web server. These scripts dynamically generate SMIL presentations which reference the media content stored on the Real Media server (the streaming video/audio components) and the web server (the text and slide images). The LecturePresenter user interface is shown in Figure 3. Users can play the video and the corresponding slides will appear automatically or they can jump to any slide and the corresponding video component will begin playing.

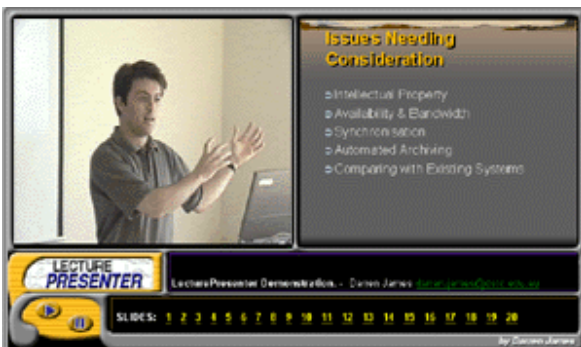


Figure 3 – An example archived seminar shown using LecturePresenter.

### 4.5 Search Interface

Currently the search interface only permits browsing by subject, lecturer and date. This is due to time restrictions and the immediate need to provide a proof of concept that synchronised presentations can be developed using SMIL. The extension of the search facilities to full-text and keyword searches as well as over the entire collection or complete lectures is seen as essential, and will be done in the near future.

## 5 Presenting with SMIL

SMIL (Synchronised Multimedia Integration Language) is basically a declarative language for describing Web-based multimedia presentations. Using a simple markup language similar to HTML, it describes how the various components of a presentation are to be combined spatially and temporally. It is a recommendation of the W3C developed by the W3C SYMM (Synchronised Multimedia) Working Group. At the time of writing this paper the 4th public Working Draft of SMIL-Boston [1] has just become available. As a neutral technology, it's objective was to solve the common problem that vendors of the rendering technologies had – how to synchronise multimedia presentations.

Because it uses a simple textual mark-up language (not unlike HTML) based on XML, it has two immediate benefits:

- Simple transition to SMIL for those familiar with HTML and/or XML
- Authoring can be done very cheaply and easily using simple text editors.

SMIL describes four fundamental aspects of a multimedia presentation:

- temporal specifications: primitives to encode the temporal structure of the application and the refinement of the (relative) start and end times of events;
- spatial specifications: primitives provided to support simple document layout;
- alternative behavior specification: primitives to express the various optional encodings within a document based on systems or user requirements;
- hypermedia support: mechanisms for linking parts of a presentation.

Essentially the SMIL mark up defines regions of a presentation (eg a video region, a slide region...) and references to the source for each region. It also allows start, end and duration times for presentation elements to be nominated. To explain this more clearly consider the following example (figure 4).

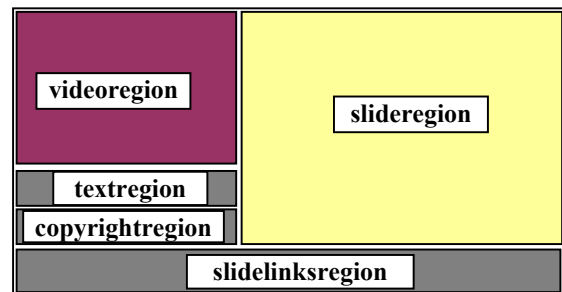


Figure 4 – The regions of an example SMIL presentation.

In Figure 4, we see clearly defined regions, and in the example SMIL file below, we can see the mark-up that

defines those regions. The SMIL file itself contains no source media for the presentation but only references to the media. The SMIL part of the presentation is relatively simple as it plays the role of director to the SMIL player.

```
<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" background-color="black"/>
    <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>
    <ref src=http://www.webserver.com.au/SMIL/Tutorial1/slides.rp
      region="slideregion"/>
    <video src="rtsp://flare.dstc.edu.au/Tutorials/Tutorial1/DFD1.rm"
      region="videoregion"/>
    <textstream src=http://www.webserver.com.au/SMIL/Tutorial1.rt
      region="textregion"/>
    <textstream
      src="http://www.webserver.com.au/SMIL/Tutorial1copyrightText.rt"
      region="copyrightregion"/>
    <textstream
      src="http://www.webserver.com.au/SMIL/Tutorial1slideLinks.rt"
      region="slidelinksregion"/>
  </par>
</body>
</smil>
```

## 5.1 SMIL Temporal Specifications

SMIL provides coarse-grain and fine-grain declarative temporal structuring of an application. Coarse grain temporal information is given in terms of two structuring elements:

- <seq> ... </seq>: A set of objects that occur in sequence.
- <par> ... </par>: A set of objects that occur in parallel.

Elements defined within a <seq> group have the semantics that a successor is guaranteed to start after the completion of a predecessor element. Elements within a <par> group have the semantics that, by default, they all start at the same time. Once started, all elements are active for the time determined by their encoding or for an explicitly defined duration.

Fine grain synchronization control is specified in each of the object references through a number of timing control relationships:

- explicit durations: a DUR=" length " attribute can be used to state the presentation time of the object;
- absolute offsets: the start time of an object can be given as an absolute offset from the start time of the enclosing structural element by using a BEGIN=" time " attribute;

- relative offsets : the start time of an object can be given in terms of the start time of another sibling object using a BEGIN=" object\_id + time " attribute.

## 5.2 The Advantages of SMIL

Below we describe the major advantages of SMIL in the context of this project.

### 5.2.1 Platform Independence

SMIL, being a W3C recommendation, and not a proprietary technology, does not tie the implementation of a synchronised presentation system to particular platforms or programming languages. While requiring a SMIL (compliant) enabled player, it does allow different client software to be plugged into the overall system without a reworking of the technology.

### 5.2.2 Synchronisation

SMIL has enabled the synchronisation of multimedia elements in a single presentation. This has been a major contributor to the success of the project. Previous projects (or commercial products) that aimed to synchronise elements involved specific technologies that were not easy to author in a flexible fashion. SMIL on the other hand is text based and easy to read.

### 5.2.3 Load Balancing

Referencing the sources (as in HTML) provides two benefits. The first being that you can change the actual sources without having to change the SMIL file<sup>1</sup>. The second benefit is that you can share the load of serving the media across different servers, even within the same presentation. For instance you may have a slow server that could be used for the image and text components and a dedicated fast server for serving up the video or audio files. The ability to change references when dynamically generating the SMIL files, is a potentially useful tool when dealing with considerably large server loads resulting from high bandwidth intensive sources (such as video) being served to many concurrent users.

### 5.2.4 Catering for Differing Connection Speeds Dynamically

SMIL incorporates a switch statement to specify a alternative options that the SMIL Player can choose from. The 'switch' can be used to dynamically configure aspects of the end user's environment such as screen size, number of colours supported by the display device, and language. But the 'switch' can also be used to determine the optimum source for media content depending on the bit rate of the download. This is an important function since many users will connect via a modem and may not have the bandwidth available to view full motion video. In this case the switch statement can be used to tell the player to make a decision

<sup>1</sup> Of course if the elements of the media's properties that affect timing or size within the presentation are affected this may have some input on the overall timing of the presentation.

based on the bit rate e.g. to use audio instead of the video or to use a lower quality video.

### **5.3 The Limitations of SMIL**

The limitations of SMIL in the context of this project are now discussed.

#### **5.3.1 Limited Support for Text**

Although SMIL mark-up can be used to reference image sources, such as gif or jpeg files, it does not provide a direct means of specifying the use of text within a presentation. For small amounts of text such as copyright notices, it is not an issue to convert the text into an image and use that in the presentation. But for large amounts of text, or text that is dynamically generated, this could be seen as a serious limitation of SMIL. Conversion of text to an image precludes the ability to perform text searches or to highlight relevant parts.

Directly marking-up text is not actually a direct function of SMIL – at least not at the highest level of the hierarchy of presentation elements. SMIL's function is to mark up where, when and for how long elements appear in a presentation. Nevertheless, the ability to mark up text is an essential requirement for presentations and SMIL does not adequately handle presenting synchronised text effectively.

#### **5.3.2 Choosing a SMIL Player**

There were eight SMIL players to choose from when the project was commenced. These are listed below:

- Apple QuickTime 4.1
- Compaq HPAS
- Helio Barbizon
- Microsoft Player Internet Explorer 5.5 (supporting selected modules of SMIL Boston draft)
- NIST S2M2 Player
- Oratrix Grins [12]
- Productivity Works L p player
- RealNetworks Realplayer 7

Of these, the most popular are Apple's QuickTime 4.1, Microsoft's IE5.5 Preview browser and RealNetworks RealPlayer.

Since the QuickTime player does now allow for rendering of text (other than a single line of limited character length in the playlist area) it was not suitable.

Microsoft's IE5.5 capability to play HTML+SMIL presentations could prove to be a very useful tool in delivering synchronised multimedia to client machines without having to download specific players. Its disadvantage is that it would reduce the possible audience to those running Windows and Internet Explorer 5.5 (which is a significant proportion of the user population).

Real Networks RealPlayer G2 has provided extensions to the SMIL mark-up that allow for text to be directly marked

up. It is essentially for this reason plus its stability and the fact that RealPlayer is available on many systems today, that it was chosen as the initial player for the system.

#### **5.3.3 Embedding within a Web Page**

SMIL does not provide support for user interaction via search term entry fields or choice boxes. This is not necessarily a shortcoming of SMIL which does the job for which it was created – to allow content creators to mark-up the elements of a presentation and to synchronise their display. However the integration of user interface capabilities with a presentation is required for this project. Although it is possible to provide this functionality via a browser running independently, additional (and possibly proprietary) technology would be required to coordinate the SMIL player and the web browser. The ideal solution is to enable the replay of SMIL presentations within normal web pages in a web browser. Internet Explorer 5.5 claims to support this functionality which will provide a significant advantage over other players and browsers.

### **5.4 Recent SMIL Advances**

Some advances have recently been made which may lead to the widespread adoption and use of SMIL. These advances all improve the integration of SMIL with HTML and allow SMIL presentations to be embedded within web pages to both enrich the web page and add user interaction functionality to SMIL presentations (eg integrated search facilities).

#### **5.4.1 HTML+Time**

HTML+Time1.0, was a joint submission by Compaq Computer Corporation, Macromedia Inc. and Microsoft Corporation, as a submission [13] to the W3C which provides a means to synchronise presentation elements within HTML. While HTML+Time1.0 has not been recommended by the W3C organisation, it supports selected modules of the SMIL-Boston draft, and provides extensions to that draft that allow the author of HTML+Time content to mark-up all elements of a presentation.

The HTML+Time1.0 submission did raise legitimate concerns about the limitations of SMIL to handle features such as simple or marked-up text and the integration with HTML. A universally accepted mark-up for web content. HTML+Time1.0 was not used in the project because of the limited platforms on which it would run (only the preview edition of Internet Explorer 5.5 running on a Windows platform is able to play HTML+Time1.0 marked-up content).

#### **5.4.2 HTML+SMIL**

The HTML+SMIL language profile[14] was a subsequent proposal to integrate a subset of the SMIL Boston specification with HTML. This subset can be used to manipulate HTML and CSS (Cascading Style Sheets) [15]. It essentially allow the migration from standard HTML (and CSS) to SMIL presentations within the same web browser. Of the SMIL Boston modules, it supports

animation, content control, linking, media objects, timing and synchronization, and transition effects.

HTML+SMIL is one of the three nominated language profiles included in the SMIL Boston draft. The other two are the SMIL Boston Language Profile and the SMIL Basic Language Profile. Language profiles allow the SMIL mark-up to use either: the base SMIL language; an extended version that allows HTML elements to be directly included; or a very basic subset that could be displayed on simple mobile devices such as hand-held computers or mobile phones.

HTML+SMIL appears to be a promising implementation which will allow SMIL presentations to be played alongside other web media (including HTML) in the same web browser. This will enable the inclusion of a textual search interface in the same web browser as the presentation – a capability which is not currently possible.

## 6 Current Applications

The system is currently being trialled by two departments within the the University of Queensland as well as the Centre for Online Health.

The School of Information Technology in conjunction with DSTC Pty Ltd is investigating incorporation of the system into their program that delivers a Masters in Information Technology degree via a flexible delivery mechanism called Flex-eL.

The Biosciences Faculty is using the system to provide supplementary material to students attending the University of Queensland's St Lucia Campus in Brisbane Australia as well as overseas students in Brunei. The trial involves around 120 hours of material given over one semester of an academic calendar.

The Centre of Online Health is trialling the system for the capture and archival of a series of seminars given by medical experts in various fields.

## 7 Conclusions and Future Work

The filming and replay of a lecture synchronised with slides does not constitute an effective teaching tool. While the standard lecture style of delivery of material is acknowledged as not the most engaging or efficient means of teaching, the playback of such lectures without opportunity for interaction serves even less use.

Videos focussing on the lecturer's face provide little added value after the initial display. Focus then shifts to the slides. However there are many situations in which focussing on the slides is not sufficient to carry the message across to the viewer. Excellent examples are concepts that are difficult for students to grasp and which are normally handled in tutorials. These tutorials will often involve the tutor presenting the material on a whiteboard or interacting with an overhead transparency to work through the problem. This situation, in which an understanding is acquired by following the discussion and rationale behind analytical steps is where a system such as the one discussed

in this paper, will prove to be invaluable. If it is integrated into the course materials as one component of a flexible delivery course rather than seen as a replacement for lecture material, then it can provide an effective educational resource. However the indexing of such unstructured discussions is much more complex than that for uninterrupted pre-planned PowerPoint presentations.

The system as it stands provides an ideal solution for the streamlined indexing and archival of presentations. Similar systems do not provide such fine-grained search, retrieval and replay of multimedia components based on subject, keywords or even full-text searches without expensive time-consuming indexing. It is this ability that makes a large collection useful as a reference or revision resource.

Future work will be investigating the development of presentation editing tools to allow updates or additions to the base presentation whilst maintaining synchronisation. This would enable educators to modify the presentation to include insert new more topical material or to provide links to resources that the student would find helpful. This is seen as essential to allow educators to provide a richer educational medium for distance education.

Future work also includes investigating the ability of Internet Explorer 5.5 to support the HTML+SMIL proposal which would allow the integration of a search facilities or additional links within the presentations. This would provide even greater flexibility to students enabling them to move through presentations according to their current needs.

## References

- [1] Synchronized Multimedia Integration Language (SMIL) Boston Specification, W3C Working Draft 22 June 2000 <http://www.w3.org/TR/smil-boston/>
- [2] XML Schema <http://www.w3.org/XML/Schema>
- [3] World Wide Web Consortium: <http://www.w3c.org>
- [4] Dublin Core Metadata Initiative <http://purl.org/dc/>
- [5] Presenter.com <http://www.presenter.com>
- [6] Webpresenter <http://www.webpresenter.net>
- [7] Mukhopadhyay, S., and Smith, Brian. Passive Capture and Structuring of Lectures. <http://www2.cs.cornell.edu/zeno/Papers/lb/lb-mm99-final.pdf>
- [8] Arts Multimedia Centre, University of Western Australia, Automated Lecture Recordings, <http://www.artsmmc.uwa.edu.au/mmc/lecture.html>
- [9] University of Texas Arlington, Center for Distance Education <http://distance.uta.edu/>
- [10] Barker L. Learning Climate and Learning: Comparing Face-to-Face and Online Classes. Conference on New Media in Higher Education and Learning. The Annenberg School for Communication and The Getty Center Los Angeles, California 27-30 October 1999

- [11] Lore XML Database. Stanford University. Available WWW: <http://www-db.stanford.edu/lore/>
- [12] D.C.A Bulterman, L. Hardman, J. Jansen, K. Sjoerd Mullender, L. Rutledge, *GriNS: A Graphical Interface for Creating and Playing SMIL Documents*. Computer Networks and ISDN Systems n30, September 1998.
- [13] HTML+Time 1.0 submission. <http://www.w3.org/Submission/1998/14/>
- [14] HTML+SMIL language profile. <http://www.w3.org/TR/2000/WD-smil-boston-20000622/html-smil-profile.html>
- [15] Cascading Style Sheets. <http://www.w3.org/Style/CSS/>
- [16] Hunter, J., Iannella, R. [The Application of Metadata Standards to Video Indexing](#). Second European Conference on Research and Advanced Technology for Digital Libraries ECDL'98, 19 - 23 September, 1998, Crete, Greece

## Appendix A: XML Schema for the Metadata Associated with a Collection of Lectures

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd=http://www.w3.org/1999/XMLSchema>

  <xsd:annotation>
    <xsd:documentation>
      Lecture Archive Schema for University Of Queensland
      Computer Science Lectures.
    </xsd:documentation>
  </xsd:annotation>

  <xsd:element name="collection" type="LectureCollection" />

  <xsd:complexType name="LectureCollection" >
    <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="Date.Created" type="xsd:string" />
    <xsd:element name="Type" type="xsd:string" />
    <xsd:element name="Format" type="xsd:string" />
    <xsd:element name="Language" type="xsd:string" />
    <xsd:element name="Rights" type="xsd:string" />
    <xsd:element name="Size" type="xsd:string" />
    <xsd:element name="lecture" type="LectureType"
      minOccurs="1" maxOccurs="unbounded"/>
    <xsd:attribute name="id" type="IDREF"/>
  </xsd:complexType>

  <xsd:complexType name="LectureType" >
    <xsd:element name="Title" type="xsd:string" />
    <xsd:element name="Description" type="xsd:string" />
    <xsd:element name="Lecturer" type="LecturerType" />
    <xsd:element name="Date" type="xsd:date" />
    <xsd:element name="Place" type="xsd:string" />
    <xsd:element name="Duration" type="xsd:timeDuration"/>
    <xsd:element name="Subject" type="SubjectType"/>
    <xsd:element name="Series" type="SeriesType"/>
    <xsd:element name="Course" type="CourseType" />
    <xsd:element name="Department" type="xsd:string" />
    <xsd:element name="Institution" type="xsd:string" />
    <xsd:element name="lectureSlides"
      type="LectureSlidesType" />
  </xsd:complexType>

  <xsd:complexType name="LecturerType" >
    <xsd:element name="Name" type="xsd:string" />
    <xsd:element name="Title" type="xsd:string" />
    <xsd:element name="email_address" type="xsd:string" />
  </xsd:complexType>

  <xsd:complexType name="CourseType" >
    <xsd:element name="CourseCode" type="xsd:string"
      minOccurs="0" maxOccurs="unbounded" />
    <xsd:element name="Description" type="xsd:string"
      minOccurs="0" maxOccurs="unbounded" />
  </xsd:complexType>

  <xsd:complexType name="SeriesType" >
    <xsd:element name="Title" type="xsd:string" />
    <xsd:element name="Description" type="xsd:string" />
  </xsd:complexType>

  <xsd:complexType name="SubjectType" >
    <xsd:element name="SubjectCode" type="xsd:string" />
    <xsd:element name="Title" type="xsd:string" />
    <xsd:element name="Description" type="xsd:string" />
  </xsd:complexType>

  <xsd:complexType name="LectureVideoType" >
    <xsd:element name="Format" type="xsd:string" />
    <xsd:element name="Transcript" type="uri" />
    <xsd:element name="Duration" type="xsd:timeDuration" />
    <xsd:element name="Rights" type="xsd:string" />
    <xsd:element name="lectureSegment"
      type="LectureSegmentType" minOccurs="1"
      maxOccurs="unbounded" />
    <xsd:element name="video" type="xsd:uriReference"
      minOccurs="0" maxOccurs="1" />
    <xsd:attribute name="id" type="IDREF"/>
  </xsd:complexType>

  <xsd:complexType name="LectureSlidesType" >
    <xsd:element name="Format" type="xsd:string" />
    <xsd:element name="Size" type="xsd:integer" />
    <xsd:element name="Rights" type="xsd:string" />
    <xsd:attribute name="id" type="IDREF"/>
  </xsd:complexType>

  <xsd:complexType name="LectureSegmentType" >
    <xsd:element name="keyword" type="xsd:string"
      minOccurs="0" maxOccurs="unbounded" />
    <xsd:element name="startTime" type="xsd:long" value="0" />
    <xsd:element name="endTime" type="xsd:long" />
    <xsd:element name="Duration" type="xsd:timeDuration" />
    <xsd:element name="slide" type="xsd:uriReference"
      minOccurs="0" maxOccurs="1" />
    <xsd:element name="text" type="xsd:string" minOccurs="0"
      maxOccurs="1" />
    <xsd:attribute name="id" type="IDREF"/>
  </xsd:complexType>
</xsd:schema>
```

```
type="LectureSlidesType">
  <xsd:element name="lectureVideo"
    type="LectureVideoType" />
  <xsd:attribute name="id" type="IDREF"/>
</xsd:complexType>

<xsd:complexType name="LecturerType" >
  <xsd:element name="Name" type="xsd:string" />
  <xsd:element name="Title" type="xsd:string" />
  <xsd:element name="email_address" type="xsd:string"/>
</xsd:complexType>

<xsd:complexType name="CourseType" >
  <xsd:element name="CourseCode" type="xsd:string"
    minOccurs="0" maxOccurs="unbounded" />
  <xsd:element name="Description" type="xsd:string"
    minOccurs="0" maxOccurs="unbounded" />
</xsd:complexType>

<xsd:complexType name="SeriesType" >
  <xsd:element name="Title" type="xsd:string" />
  <xsd:element name="Description" type="xsd:string" />
</xsd:complexType>

<xsd:complexType name="SubjectType" >
  <xsd:element name="SubjectCode" type="xsd:string" />
  <xsd:element name="Title" type="xsd:string" />
  <xsd:element name="Description" type="xsd:string" />
</xsd:complexType>

<xsd:complexType name="LectureVideoType" >
  <xsd:element name="Format" type="xsd:string" />
  <xsd:element name="Transcript" type="uri" />
  <xsd:element name="Duration" type="xsd:timeDuration" />
  <xsd:element name="Rights" type="xsd:string" />
  <xsd:element name="lectureSegment"
    type="LectureSegmentType" minOccurs="1"
    maxOccurs="unbounded" />
  <xsd:element name="video" type="xsd:uriReference"
    minOccurs="0" maxOccurs="1" />
  <xsd:attribute name="id" type="IDREF"/>
</xsd:complexType>

<xsd:complexType name="LectureSlidesType" >
  <xsd:element name="Format" type="xsd:string" />
  <xsd:element name="Size" type="xsd:integer" />
  <xsd:element name="Rights" type="xsd:string" />
  <xsd:attribute name="id" type="IDREF"/>
</xsd:complexType>

<xsd:complexType name="LectureSegmentType" >
  <xsd:element name="keyword" type="xsd:string"
    minOccurs="0" maxOccurs="unbounded" />
  <xsd:element name="startTime" type="xsd:long" value="0" />
  <xsd:element name="endTime" type="xsd:long" />
  <xsd:element name="Duration" type="xsd:timeDuration" />
  <xsd:element name="slide" type="xsd:uriReference"
    minOccurs="0" maxOccurs="1" />
  <xsd:element name="text" type="xsd:string" minOccurs="0"
    maxOccurs="1" />
  <xsd:attribute name="id" type="IDREF"/>
</xsd:complexType>
</xsd:schema>
```



## Appendix B: Examples of Instantiated XML Descriptions of a Collection and Lecture

```
<collection id="CSEEDepartmentLectures">
  <Title>University of Qld. CSEEDepartmentLectures</Title>
  <Creator>Darren James</Creator>
  <Subject>University Lectures</Subject>
  <Description>
    A collection of lectures that have been given at the
    University of Queensland for subjects or series held by
    the CSEE Department.
  </Description>
  <Publisher>DSTC Pty Ltd</Publisher>
  <Date_Created>2000-06-14</Date_Created>
  <Type>Computer Science Lecture Material</Type>
  <Format>SMIL</Format>
  <Language>en</Language>
  <Rights>University of Queensland</Rights>
  <Size>123</Size>
</collection>
```

```
<lecture id="MInfTech-01/00-Lecture3">
  <Title>Data Flow Diagrams</Title>
  <Description>An Introduction to Data Flow Diagrams
</Description>
  <Lecturer>
    <Name>Darren James</Name>
    <Institution>Distributed Systems Technology
      Centre</Institution>
    <email_Address>darren.james@dstc.edu.au
    </email_Address>
  </Lecturer>
  <Audience> Masters of Information Technology Students
</Audience >
  <Place>University of Queensland St Lucia</Place>
  <Date>2000-07-10</Date>
  <Duration>1 hour</Duration>
  <Subject>
    <SubjectCode>CS834</SubjectCode>
    <Title>Project A</Title>
    <Description>M.Inf.Tech Project Part A</Description>
  </Subject>
  <Course>
    <CourseCode>M.InfTech</CourseCode>
    <Description>Masters of Information Technology
    </Description>
  </Course>
  <Department>Computer Science and Electrical Engineering
</Department>
  <Institution>University of Queensland</Institution>
</lecture>
```

```
<lectureSlides id="http://uq.csee.edu.au/dxj/lecture3.ppt"
  <Format>PowerPoint 2000</Format>
  <Size>15</Size>
  <Rights>Darren James</Rights>
</lectureSlides>
```

```
<lectureVideo id="http://uq.csee.edu.au/dxj/lecture3.rm">
  <Format>RealVideo 8</Format>
  <Duration>56min</Duration>
  <Rights>University of Queensland, DSTC</Rights>
  <lectureSegment id="1">
    <keyword>DFD</keyword>
    <keyword>Data</keyword>
    <keyword>Flow</keyword>
    <keyword>Diagram</keyword>
    <startTime>0:0:0</startTime>
    <endTime>0:0:17</endTime>
    <Duration>0:0:17</Duration>
    <text>What is a DFD Data Flow Diagram Useful for
    showing where data is flowing to and from in a
```

```
    system</text>
  </lectureSegment>
  <lectureSegment id="2">
    <keyword>Benefits </keyword>
    <keyword>Objectives</keyword>
    <keyword>DFD</keyword>
    <startTime>0:0:17</startTime>
    <endTime>0:0:48</endTime>
    <Duration>0:0:31</Duration>
    <text>Why do we bother? We need to understand
    what components of the implementation require
    specific information</text>
  </lectureSegment>
```

```
...
</lectureVideo>
```