Multimedia Systems (2005) 10(6): 544–558
DOI 10.1007/s00530-005-0181-8

REGULAR PAPER

Silvia Pfeiffer · Conrad Parker · André Pang

The Continuous Media Web: a distributed multimedia information retrieval architecture extending the World Wide Web

Published online: 8 August 2005

Abstract The World Wide Web, with its paradigms of surfing and searching for information, has become the predominant system for computer-based information retrieval. Media resources, however information-rich, only play a minor role in providing information to Web users. While bandwidth (or the lack thereof) may be an excuse for this situation, the lack of surfing and searching capabilities on media resources are the real issue. We present an architecture that extends the Web to media, enabling existing Web infrastructures to provide seamless search and hyperlink capabilities for time-continuous Web resources, with only minor extensions. This makes the Web a true distributed information system for multimedia data. The article provides an overview of the specifications that have been developed and submitted to the IETF for standardization. It also presents experimental results with prototype applications.

Keywords Continuous Media Web · Annodex · CMML · Markup of media · Open media metadata standards

1 Introduction

Since the Web's very foundation by Tim Berners-Lee, academics, entrepreneurs, and users have dreamt of ways to integrate the Web's various multimedia contents into one distributed system for storage *and* retrieval. The power of the World Wide Web stems from being a *world-wide distributed information storage system* architected to be scalable, simple to author for, and simple to use. Information is found mostly through Web search engines such as Google or through Web portals, both of which commonly just provide starting points for hyperlinking (*surfing*) to other related documents. Information nowadays encompasses heterogeneous Web resources of all sorts, including media. The uniform means of accessing information on the Web is through hyperlinks given in the URI (Unified Resource Identifier) format (see `http://www.w3.org/Addressing/`).

The World Wide Web is currently the predominant distributed information retrieval architecture on the Internet [2]. However, it is not yet a distributed information retrieval system for *time-continuously sampled data* such as audio and video. It is only set up well for information retrieval on HTML resources. Although the number of audio and video data on the Internet is increasing, these documents cannot be used as intensively as HTML content.

While URIs provide a uniform means to access and explore information, they can only point to a complete audio or video file, not into a specific temporal offset or a specific segment of interest. Also, when viewing an audio or video resource, the hyperlinking functionality of the Web is "left behind" and a dead end is reached. Audio or video resources cannot typically hyperlink to further Web resources, thus interrupting the uniform means of information exploration provided by the Web.

Web information search is also deficient for media resources. While it is nowadays possible to search the content of binary text documents, such as Postscript files or word processor documents, and identify whether they contain information relevant to a particular query, this is not possible for media resources. Media resources do not currently have a textual representation that fits into the text-based indexing paradigm of existing Web search engines, and therefore their content, however information-rich, cannot be searched uniformly on the Web.

Realizing this situation, we have developed an extension to the World Wide Web that addresses these issues. We are proposing a new standard for the way time-continuously sampled data such as audio and video files are placed on Web sites to make media content just as "surfable" as ordinary text files. In this new standard, users can link not only from a text passage to e.g., a video, but into a specific time interval containing the information being sought [21]. The time-continuous resource itself is annotated with HTML-like markup and has an XML representation of its content, enabling Web search engines to index it. The markup may

S. Pfeiffer (✉) · C. Parker · A. Pang
CSIRO-ICT Centre PO Box 76 Epping, NSW 1710, Australia
E-mail: {Silvia.Pfeiffer, Conrad.Parker, Andre.Pang}@csiro.au

contain hyperlinks to other Web resources, which enables search engines to crawl time-continuous resources in the same way as they crawl HTML pages. These hyperlinks also enable the Web user to hyperlink through a time-continuous resource, thus integrating these resources into the "surfing" paradigm of the Web.

At CSIRO we call the Web that is extended to time-continuous resources the *Continuous Media Web (CMWeb)* [24]. The technology is based upon the observation that the data stream of existing time-continuous data might be temporally subdivided based on a semantic concept, and that this structure enables access to interesting subparts, known as *clips*, of the stream. This structure is captured in an HTML-like markup language, which we have named *Continuous Media Markup Language (CMML)* [23]. CMML allows annotations and hyperlinks to be attached to media resources. For synchronized delivery of the markup and the time-continuous data over the Web, we have developed a streamable container format called *Annodex* [22], which encapsulates CMML together with the time-continuous resource. When consuming (i.e., listening, viewing, reading) such an annotated and indexed resource, the user experience is such that while, e.g., watching a video, the annotations and links change over time and enable browsing of collections simply by following a link to another Web resource.

The remainder of this article is organized as follows. Section 2 provides some definitions for the key concepts discussed in this article, and Sect. 3 discusses some related research work. Section 4 describes the specifications of the CMWeb architecture and technologies, introducing temporal URI addressing, the Continuous Media Markup Language (CMML), and the Annodex format. Switching the focus to the actual handling of Annodex resources, Sects. 5 and 6 detail how an Annodex resource is being produced from an existing media file such as an *MPEG-1* encoded video and how it is distributed over the Internet. Section 7 looks at information retrieval through Web search engines. Section 8 provides some experimental results of first implementations. Section 9 compares the CMWeb to other media standards with respect to their capabilities of extending the World Wide Web to a distributed multimedia information retrieval infrastructure. Finally, Sect. 10 concludes the paper with a summary and outlook for future work.

2 Definitions

Before we can discuss work in this field, we need to establish that the terms we use have a well-understood meaning. Therefore, we now introduce some fundamental definitions.

- *Time-continuously sampled data* are any sequence of numbers that represents an analog-time signal sampled in discrete time steps. In contrast to actual discrete-time signals as known from signal processing, time-continuously sampled data may also come in compressed form, such that a block of numbers represents an interval of time. Decompression will lead to the higher temporal resolution that the resource was sampled at in the first place. This is the case for all common audio and video compression formats. Other types of time-continuously sampled data may, for example, be physical measurements of natural phenomena, such as the acidity (pH) value of a lake, or the temporally changing value of a stock market. Time-continuously sampled data are also sometimes just called *time-continuous data* or, in the case of audio or video, just *media*.
- *Time-continuous (Web) resource* is a time-continuous data stream that can be distributed by a Web server. The resource may exist as a file or may be created on the fly (live or composed). To retain the ability to handle and display the time-continuous data progressively during delivery over HTTP, a time-continuous Web resource must further be streamable, i.e., it must be possible to decode it in a single linear pass using a fixed amount of memory.
- *Clip* is an arbitrary temporal section of a time-continuous Web resource as defined by a user. A clip makes sense with respect to some semantic measure of the user. A whole time-continuous Web resource may be regarded as a clip, though the more common case is to describe such a resource as a combination of several clips.
- *Annotation* is a fee-text, unstructured description of a clip.
- *Metadata* are a set of name–value pairs that provide databaselike structured descriptions, e.g., in HTML metadata are represented in meta elements within the head tag [29]. An existing metadata scheme that is independent of the format of data that it can describe is the Dublin Core [7]).
- *Hyperlink* is a Unified Resource Identifier that can point to or into any Web resource. In this context, the term *media browsing* is then mainly used for following hyperlinks into and out of media files rather than looking through a collection of media files, as is the more common use of the term in media management systems.
- *Meta-information* is a collection of information about a time-continuous data stream, which may include annotations, metadata, and hyperlinks.
- *Markup* is the collection of tags that are used to provide metainformation to a Web resource. The most common language for providing markup is XML [33].

3 Related research

There have been several approaches to a better integration of media documents into the World Wide Web that have not been standards based. These will be briefly discussed here. A standards-based approach is superior as it enables interoperability between different proprietary solutions and scales over the whole Internet rather than providing a solution limited to a subpart of the Internet only.

The database community has been addressing this issue. Here, the idea is to use databases to store references to media objects and metainformation about them (see, e.g., [14, 16]).

This approach requires the introduction of a middle layer (middleware) to provide an interface between a Web server and the database in the backend. A media presentation, such as a SMIL presentation, can be dynamically created by the middleware upon a user's HTTP request. Such hiding of information in databases, however, excludes Web search engines from accessing the metainformation about the media objects. Also, hyperlinking into clips of media is not generally possible. Therefore, while the database approach solves content management requirements for time-continuous Web resources, it does not extend the searching and surfing infrastructure of the Web to audio or video. Most of the approaches in the database community rely on SMIL do provide that functionality. We will discuss the feasibility of SMIL later in this article.

The search engine community has approached the partial problem of searching media on the Web. While crawling for common audio, video, or image content is not possible as these resources do not have hyperlinks associated with them, this community has come up with interesting special solutions, most of which are based on simple metainformation attached to the data via filenames or simple schemes for specific resource types such as ID3 tags [19] for MP3 files. A good collection of currently available media search engines can be found at `http://searchenginewatch.com/links-/article.php/2156251` [28]. These are special solutions, partly also resulting from signal-processing-based automated audio and video content analysis techniques developed over the last decade (see, e.g., [11, 17, 27, 37]). Nobody has as yet been able to handle time-continuous Web resources and HTML Web pages in a single query entry interface.

4 The CMWeb technology

To turn the World Wide Web into an architecture for multimedia information retrieval, it is instructional to examine what turned the Web into an architecture for text information retrieval. This will help extrapolate the gaps that need to be filled to extend the Web's distributed information retrieval infrastructure to time-continuous data.

The World Wide Web was created by three core technologies [4]: HTML, HTTP, and URIs. They respectively enable:

- The markup of textual data, giving them an addressable structure, metadata, and outgoing hyperlinks;
- The distribution of the marked-up documents over a simple Internet-based protocol providing a standard interchange format; and
- The hyperlinking to and into Web documents.

These together created the scalable distributed information retrieval architecture we know as the World Wide Web with its uniform means of accessing, exploring, and searching for information. These core technologies enabled the implementation of Web browsers and Web search engines, which turned the Web into a distributed information retrieval system. However, time-continuous data such as audio or video currently do not form part of the Web's searching and surfing infrastructure. What are they missing?

Extrapolating from the experiences of the Web, the following three core capabilities are required to enable a Web of time-continuous resources:

- A markup language to create an addressable structure, searchable metadata, and outgoing hyperlinks for a time-continuous resource;
- An integrated document format that can be distributed via HTTP and enables synchronized delivery of markup and data, providing a standard interchange format; and
- A means to hyperlink into time-continuous resources.

The technical challenge for the development of the CMWeb [1] was the creation of a solution to these three issues. We have therefore developed three specifications:

1. CMML [23], the Continuous Media Markup Language, which is based on XML and provides tags to mark up time-continuous data into sets of annotated temporal clips. CMML draws upon many features of HTML [29].
2. Annodex [22], the binary stream format to store and transmit interleaved CMML and media data.
3. Temporal URIs [21], which enable hyperlinking to temporally specified sections of an Annodex resource.

The cited documents are works in progress, and changes may be made as the technical discussions with standards bodies and other experts continue. For the latest versions, check out `http://www.annodex.net/specifications.html`. Reference implementations of the core Annodex technology, including desktop browsers, for the CMWeb have been developed and are available at `http://www.annodex.net`.

The following three subsections present in more detail the three core technologies that enable the CMWeb. See Sect. 8 for example implementations of a CMWeb browser, an Annodex-enabled search engine, and an Annodex enabled Web server.

4.1 Temporal URI references

Web-integrated access to clips or temporal offsets in time-continuous data requires URIs [3] that point to such resource references. Already [25] discussed the need to specify better addressing schemes for media fragments to improve integration of media content into the Web.

We have developed two ways to point to subparts of time-continuous data: URI fragment identifiers and URI query components. According to the URI standard [3] *URI fragments* are specified in a URI after the hash # character and *URI queries* after the question mark? character.

By definition, URI fragments can be interpreted on the client application only [3]. However, media data are usually high-bandwidth and large-size data. Hence, downloading a complete media file before performing the offset action may not be desirable if the user has to wait for an unacceptable amount of time. Therefore, the same scheme that is used to access fragments locally is also proposed as a generic URI query scheme to tell the server to provide only the requested fragment(s) of the media.

Our proposed URI format for fragment and query identifiers for time-continuous Web resources is specified in an Internet Draft [21] and is conformant to the URI standard given in RFC 2396 [3]. The syntax is closely related to the specification of relative timestamps of the RTSP protocol parameters as given in RFC 2326 [26].

Two fundamentally different ways of addressing information in an Annodex resource are necessary: addressing of clips and addressing of time offsets or time segments. These ways of addressing also extend to CMML resources, as these are essentially textual representations of the content of an Annodex resource. Note that as CMML resources are XML files (Sect. 4.2), any element in a CMML resource can also be addressed using XPointer [32] and XPath [35] constructs. These addressing schemes do not extend to binary data such as time-continuous data, and the schemes provided here were developed to bridge that gap.

4.1.1 Addressing of clips

Clips in Annodex resources (file extension `.anx`) are identified by their *id* attribute. These *id* values create anchor points to which URI references can be attached. Thus, accessing a named clip in an Annodex resource is achieved with the following CGI [18] conformant query parameter specification:
id="clip_id"
or the following fragment specification:
#clip_id

Examples for accessing a clip in an example CMML or Annodex resource are:

http://foo.bar/example.cmml?id="findingGalaxies"
http://foo.bar/example.anx?id="findingGalaxies"
http://foo.bar/example.cmml#findingGalaxies
http://foo.bar/example.anx#findingGalaxies

For delivering the queried CMML or Annodex resources from a CMWeb Server, the resource may need to be processed before being served out. To gain a conformant data format that starts at the clip offset, the inherent basetime of the resource has to be adjusted to the start time of the requested clip, which is only possible for a format that, like Annodex, stores a basetime for the first frame of its encapsulated time-continuous data. Then, an interval of the resource has to be extracted starting with the time-continuous data from the queried clip data, which needs to be appended to adjusted file headers.

4.1.2 Linking to time segments

It is also desirable to be able to address any arbitrary time segment of an Annodex or CMML resource. This is again achieved with a CGI conformant query parameter specification:

`t=[time-scheme:]time_interval`

or the following fragment specification:

`#[time-scheme:]time_interval`

Available time schemes are *npt* for normal play time, *smpte* specifications of the Society of Motion Pictures and Television Engineers (SMPTE), and *clock* for a Universal Time Code (UTC) time specification. For more details, see the specification document [21].

Examples for requesting one or several time intervals from a given sample CMML and Annodex resources are:
http://foo.bar/example.cmml?t=85.28
http://foo.bar/example.cmml#85.28

http://foo.bar/example.anx?t=npt:15.6-85.28, 100.2
http://foo.bar/example.anx#npt:15.6-85.28,100.2
http://foo.bar/example.anx?t=smpte-25:00:01: 25:07
http://foo.bar/example.anx#smpte-25:00:01:25:07

.../example.anx?t=clock:20040114T153045.25Z
.../example.anx#clock:20040114T153045.25Z

Where only a single time point is given, this is interpreted to relate to the time interval covered from that time point onward until the end of the stream.

The same preprocessing as described in the previous subsection is required for these specifications.

4.2 CMML

The Continuous Media Markup Language [23] has been designed to cater for two different yet related uses:

- Authoring of clip structures, anchor points, annotations, and hyperlinks for time-continuous data in preparation for integration with the data in an Annodex resource.
- Indexing and crawling of time-continuous Web resources for search engines to perform retrieval on a textual representation of the binary Annodex resources.

CMML is an XML-based language to describe the content of a time-continuous Web resource. It is an authoring language for annotating, indexing, and hyperlinking time-continuous data in Annodex format. A CMML file contains structured XML markup where we have chosen the XML tags to be very similar to XHTML to enable a simple transfer of knowledge for HTML authors.

CMML documents consist of three main types of tags: at most one *stream* tag, exactly one *head* tag, and an arbitrary number of *clip* tags. The *stream* tag is optional and describes the input bitstream(s) necessary for the creation of

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE cmml SYSTEM "cmml.dtd">

<cmml>

<stream basetime="npt:0">
  <import id="a1" contenttype="video/mpeg"
          src="index.mpg" start="npt:0"/>
</stream>

<head>
  <title>The Research Hunter</title>
  <meta name="Producer" content="CSIRO"/>
</head>

<clip id="intro" start="npt:0">
  <meta name="Location" content="Pond at Uni"/>
  <meta name="Subject" content="Intro to CSIRO"/>
  <a href="http://www.csiro.au/">
     Link to CSIRO homepage
  </a>
  <img src="/media_images/index1.jpg"/>
  <desc>
     Welcome to CSIRO, the Commonwealth Scientific
     and Industrial Research Organisation of
     Australia.
  </desc>
</clip>

<clip id="astronomy" start="npt:33">
  <a href="00-4-5.anx#startspeech">
     Tidbinbilla Honeysuckle space tracking station
  </a>
  <img src="/media_images/index2.jpg"/>
  <desc>
     Our astronomers track the Galileo space
     probing station in Tidbinbilla.
  </desc>
</clip>

<clip ...
</clip>

</cmml>
```

Fig. 1 Extract of a CMML file with stream, head, and clip tags

an Annodex resource. The *head* tag contains information related to the complete time-continuous resource. A *clip* tag, in contrast, contains information on a temporal fragment of the data.

Figure 1 shows an example of a CMML document. It describes two clips for an MPEG video about the "Research Hunter." The media file is referred to in the *stream* tag as "index.mpg," and the title in the *head* tag provides the subject, while details about the clips are found in the *clip* tags. This is an actual example from our test collection at `http://media.annodex.net/sciweb/`.

The XML markup of a *head* tag in the CMML document contains information on the complete time-continuous Web resource. Its essential information contains:

- Structured descriptions in *meta* tags and
- Unstructured textual annotations in the *title* tag.

The XML markup of a clip tag contains information on a fragment of time-continuous data:

- Anchor points (i.e., the *id* attribute) to provide locations inside a time-continuous resource that a URI can refer to. Anchor points identify the start of a clip. This enables URIs to refer to clips in CMML or Annodex resources via fragment specifications as described in Sect. 4.1.
- Structured textual annotations in the *meta* tags in the same way as the head tag.
- Unstructured textual annotations in the *desc* tags. Unstructured annotations are free text and mainly relevant for search applications.
- An optional keyframe in the img tag providing a representative image for the clip and enabling display of a table of contents for Annodex resources.
- Outgoing URI links in the *a* tag of the clip can point to any other place a URI can point to, such as clips in other Annodex resources or anchors in HTML pages.

4.3 Annodex format

Annodex is the format in which media with interspersed CMML markup is exchanged. Analogous to a normal Web server offering a collection of HTML pages to clients, an Annodex server offers a collection of Annodex resources. After a Web client has issued a URI request for an Annodex resource, the Web server delivers the Annodex resource, or an appropriate subpart of it according to the URI query parameters.

Annodex resources conceptually consist of one or more media streams and one CMML annotation stream, interleaved in a temporally synchronized way. The annotation stream may contain several sets of clips that provide alternative markup tracks for the Annodex resource. This is implemented in CMML through a *track* attribute of the *clip* tag. The media streams may be complementary, such as an audio track with a video track, or alternative, such as two speech tracks in different languages. Figure 2 shows a conceptual representation of an example Annodex resource with three media tracks (light colored bars) and two annotation tracks (darker clips) with a header describing the complete resource (dark bar at the start).

The Annodex format enables encapsulation of any type of streamable time-continuous data and is thus independent

Fig. 2 Conceptually represented example Annodex file, time increasing from *left* to *right*

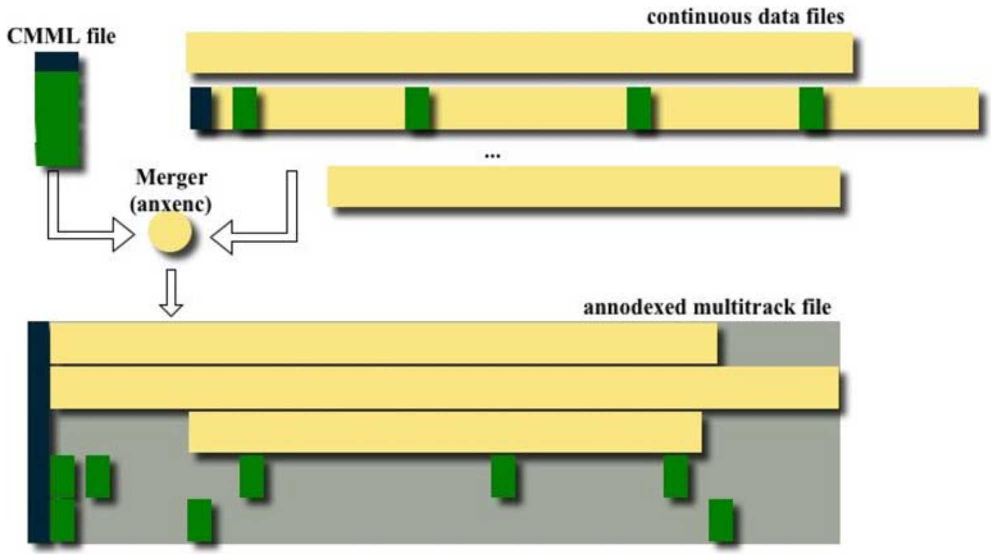

Fig. 3 Merging of the frames of several media bitstreams with a structured CMML file into an Annodexed bitstream

of a media compression format. It is basically a bitstream consisting of continuous media data interspersed with the structured XML markup of the CMML file. This is performed by merging the *clip* tags time-synchronously with the time-continuous bitstreams on authoring an Annodex bitstream. The *clip* tags are regarded as state changes in this respect and are valid from the time that they appear in the bitstream until another *clip* tag replaces them. If there is no *clip* that directly replaces a previous one, an empty *clip* tag is inserted that simply marks the end of the previous *clip* tag. Thus, Annodex is designed to be used as both a persistent file format and a streaming format.

Figure 3 shows an example of the creation of a bitstream of an Annodexed media resource. Conceptually, the media bitstreams and the annotation bitstreams share a common timeline. When encapsulated into one binary bitstream, these data have to be flattened (serialized). CMML is designed for serialization through multiplexing. The figure shows roughly how this is performed.

There are several advantages to having an integrated bitstream that includes the annotations in a time-synchronous manner with the media data. Firstly, all the information required is contained within one resource that can be distributed more easily. Also, many synchronization problems that occur with other media markup formats such as SMIL [31] are inherently solved. Also, when extracting temporal intervals from the resource for reuse, the metainformation is included in the media data, which enables one, e.g., to retain the copyright information of a clip over the whole lifetime of a reused clip. Last but not least, having a flat integrated format solves the problem of making the Annodex resource streamable.

To perform the encapsulation, a specific bitstream format was required. As stated, an Annodex bitstream consists of XML markup in the annotation bitstream interleaved with the related media frames of the media bitstreams into a single bitstream. It is not possible to use straight XML as encapsulation because XML cannot enclose binary data unless encoded as Unicode, which would introduce too much overhead. Therefore, an encapsulation format that could handle binary bitstreams and textual frames was required.

The following list gives a summary of the requirements for the Annodex format bitstream:

- Framing for binary time-continuous data and XML.
- Temporal synchronization between XML and time-continuous media bitstreams.
- Temporal resynchronization after parsing error.
- Detection of corruption.
- Seeking landmarks for direct random access.
- Streaming capability (i.e., the information required to parse and decode a bitstream part is available at the time at which the bitstream part is reached and does not come, e.g., at the end of the stream).
- Small overhead.
- Simple interleaving format with a track paradigm.

We selected Xiph.Org's [36] Ogg encapsulation format version 0 [20] as the encapsulation format for Annodex bitstreams as it meets all the requirements, has proven reliable and stable, and is an open IETF (Internet Engineering Task Force, `http://www.ietf.org/`) standard [20]. Hierarchical formats like MPEG-4 or QuickTime were deemed less suitable as they are hierarchical file formats and therefore could not easily provide for streamable, time-accurate interleaving of multiple media and annotation tracks.

5 Authoring

To author Annodexed media, we must distinguish between files and live streams. The advantage of the former is that a file can be uploaded from the computer's file system and annotated in a conventional authoring application. In contrast,

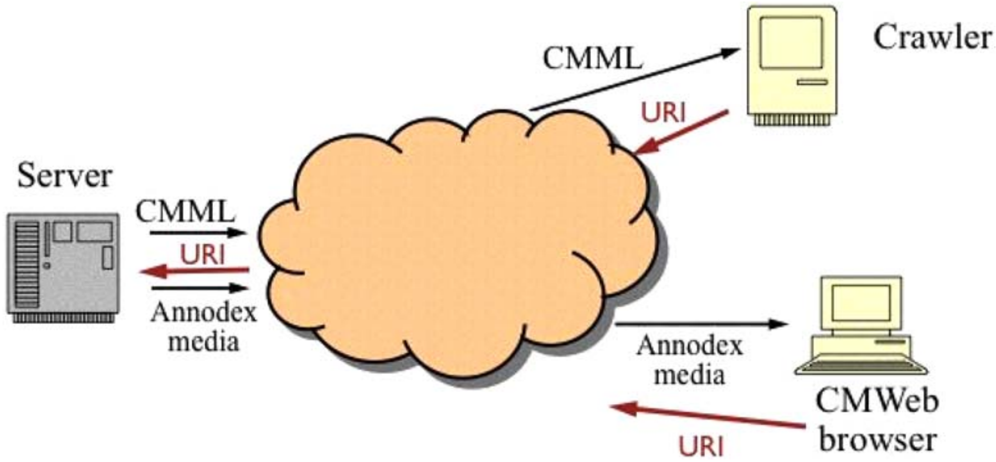

Fig. 4 Network view of the Continuous Media Web architecture

the markup of a live Internet stream by its very nature has to be done on the fly.

Annodex media files may be created in a traditional authoring application (e.g., iMovie or Adobe Premiere may easily support Annodex in the future) or through the use of CMML transcoded from metainformation collected in databases. The authoring application should support the creation of:

- Structured and unstructured annotations,
- Keyframe references,
- Anchor points, and
- URI links for media clips.

Live Annodexed media streams must be created by merging clip tags with the live digital media stream. A merger application, similar to that described in Fig. 3, inserts clip tags into the live stream at any point in time under the control of a user, e.g., by selecting a previously prepared clip tag from a list.

It is expected that extending existing graphical video editing applications such as Apple's iMovie or Adobe's Premiere to author Annodex will be a simple task. Most already provide for specific markup to be attached to fragments of a video (sometimes also called *chapters* or *regions*), thus extending the set of metainformation to cover keyframes, annotations, and hyperlinks should be fairly straightforward. The availability of such tools will support the uptake of Annodex for the average Web user, though computer specialists can already author Annodex by using anxenc and CMML.

6 Distribution

The distribution of Annodex resources over the Internet is based on URIs, similar to the distribution of HTML pages for the World Wide Web. Annodex resources are basically accessible via any of the protocols currently used to transport media formats, e.g., RTP/RTSP [26] or HTTP [8]. The use of Annodex over RTP requires the definition of a payload format for Annodex, which is future work. So far we have been using Annodex only over HTTP.

The basic process for the distribution and delivery of an Annodex resource is the following. A client dispatches a download or streaming request to the server with the specification of a certain URI. The server resolves the URI and starts packetizing an Annodexed media document from the requested clip or time, issuing a head tag at the start.

As an alternative to streaming/downloading Annodexed media from a URI, we also envisage that different applications may prefer to retrieve only either the continuous media data or the CMML transcription. Examples are browsers that cannot handle the XML markup, and information collection applications such as search engines that do not require the media data, but just its textual representation in CMML. This is possible via the `content-type` flag in an HTTP client request (Fig. 4).

7 Information retrieval

For exploiting the rich metainformation provided by Annodex resources about media, a special media player or browser plugin is necessary. Such an application has to split an Annodex resource into its constituent header and clip tags, and the media data (the reverse of the process specified in Fig. 3). A decoder is required for the given media encoding format to display the underlying media data. While playing back the media data, the application should display the hyperlinks and the annotations for the active clip. If the displayed media data is a file and not a live stream, it is even possible to display a table of contents extracted from the annotations of the file and browse through the file based on that. The hyperlinks allow the user to freely link back and forth between Annodexed media clips, HTML pages, and other Web resources. This is transparent to the user, i.e., the user "surfs" the Annodex media resources in the way

Fig. 5 Screenshot of a CMWeb browser

familiar from browsing the Web, because Annodex media seamlessly integrate with the existing Web.

Search engines can include Annodex media resources into their search repertoire effortlessly because CMML is very similar to HTML and thus implementation of parsing support for CMML is straightforward. Indexing is performed on a per-clip basis. A search engine finds annotations in the clip tags in the desc and metatags, independent of the encoding format of the media data encapsulated in Annodex. For crawling Web resources, the search engine uses the hyperlinks given in the *a* element in the *clip* tag. Thus both indexing and crawling are supported easily with Annodex. In addition, the HTTP protocol allows one to download only the CMML markup of a published Annodexed media resource by setting in the HTTP request the `Accept` header with a higher priority on the media type `text/x-cmml` than on `application/x-annodex`. A conformant Annodex server will then only provide the extracted CMML content for the given Annodex resource. This prevents crawlers from creating extensive network loads. It also reduces the size of search archives, even for large amounts of published Annodex resources, because a CMML resource contains all searchable annotations for the media clips of its Annodex resource.

8 Experimental results

To accumulate experience with the three developed technologies, we implemented several applications. The core applications required are a CMWeb server, a CMWeb browser, and a means to author Annodex resources. Then, a set of Annodexed audio and video content can be created and experiments on CMWeb browsing undertaken. A CMWeb search engine is also required to allow for information retrieval to occur. We implemented all of these core applications. In this section, we report on our experiences with these applications.

A *CMWeb server* is best implemented by extending an existing Web server with some Annodex-specific functionality. We implemented one by extending the Apache Web server [13] with a module that controls the distribution of Annodex content. It has in particular the following functionality:

1. Parses the HTTP `Accept` header to return the correct content type (either Annodex or CMML content).
2. Parses the URI of the HTTP request to identify a potential query component and return the requested subpart of the Annodex resource.
3. Checks if the requested Annodex resource exists on the server or has to be created on the fly from a CMML file by merging with the media stream(s) referred to in the stream tag.

As we implemented most of the required authoring functionality in libraries, the Apache Annodex module `mod_annodex` has turned out to be a fairly small piece of code with only about 500 lines of code. It is being used on a production Apache server (`http://media.annodex.net/`) that serves Annodex or CMML content reliably.

A *CMWeb browser* is a CMWeb client that can display Annodexed audio and video streams coming from a CMWeb server and provides a rich functionality for interaction with the content as enabled through the CMML tags. The use of plugins for existing Web browsers is our ultimate goal such that we can enable a full integration of Annodex content with existing Web content and retain a common browsing history. Our first prototype implementation is, however, a standalone CMWeb browser, which can be used as an external helper application to existing Web browsers. Figure 5 shows a screenshot of this prototype.

The various features of this CMWeb browser include:

1. *Media player*: Display and transport control of the video or audio.
2. *Browser*: History of the browser (i.e., back and forward), reload, and stop buttons as well as URI of the resource on display.
3. *Table of contents*: List of the clips making up the current Annodex resource, including representative keyframes, timestamps, and short descriptions. id tags in CMML clips allow for this list of clips to be created and to be pointed to directly. The keyframes come from img tags and the text from title tags of CMML clips.
4. *Annotation*: Additional free-text information for the current clip stored in the desc tag in CMML clips.
5. *Metadata*: Additional structured text information for the current clip stored in the metatags of CMML clips.
6. *Hyperlink*: Attached to the current clip, a clip-dependent (i.e., time-sensitive) hyperlink points to other Web resources including other Annodex resources. It comes from the *a* tag of the CMML clip.

The browsing interface of this CMWeb browser has proven a very successful design. Once the concepts of browsing webs of audio and video are understood, people find it intuitive to use the CMWeb browser to follow hyperlinks, go back and forward in history, get an overview of the current resource through the table of contents, and go directly to clips of interest.

To allow for *authoring of CMWeb content*, we implemented two means of creating Annodex content from CMML files: a standalone command-line Annodex encoding tool anxenc and a dynamic on-the-fly authoring capability of the Annodex Apache module `mod_annodex`. Both use the same libraries for parsing of CMML, audio, and video files and merging them into an Annodex bitstream. CMML files have so far been created manually using common text editors. There is potential to automate some of the markup creation task through, e.g., automated speech recognition and automated audiovisual content analysis tools or through transcoding of existing meta information to CMML.

Several *CMWeb content sites* have been created to demonstrate the CMWeb concepts, experiment with the usability of the technology, and prove that it provides the necessary data to allow a Web search engine to retrieve relevant clips for a query. In particular we have created content sites with:

- *Interviews* with scientists and video reports on results of scientific research from the CSIRO (Commonwealth Scientific and Industrial Research Organization). This site contains 29 audio and video files with 70 clips, and the CMML content has been created manually.
- *News* content from the Australian Broadcasting Corporation. This site contains 6 files with 41 clips, and the CMML content has been created mostly automatically through an xsl transform script [30] from metainformation available in NewsML [10].
- *Movie* content from the Australian Associate Press. This site contains 6 + 23 video files, and the CMML content has been created manually.
- *financial news* content from the Australian Associate Press. This site contains 5 Annodex video files, and again the CMML content has been created manually.

In summary, more than 69 Annodex video and audio files have been created as trial material covering more than 3 hours material. While the manual creation of CMML files for these is a labor-intensive process, we have been able to prove that automation is possible, especially where preexisting structured metainformation is available that can be transformed with simple text-parsing programs.

The information retrieval experience that the CMWeb creates is only complete with a search application. We therefore extended an existing *search engine* with the ability to index and crawl Annodex and CMML files. Figure 6 shows a screenshot of an example query result on the science content site.

The tested *Annodex search engine* is CSIRO's Panoptic search engine (see `http://www.panopticsearch.com/`), an ordinary Web search engine extended with the ability to crawl and index the CMML markup. As the search engine already allowed for indexing and crawling of XML resources, extension to CMML was straightforward to allow it to search and retrieve CMWeb content.

The search engine retrieves clips that are relevant to the user's query using its existing algorithms and presents ranked search results based on the relevance of the markup of the clips. A keyframe for the retrieved clips and their description are displayed. Display of such a thumbnail has been identified in [12] as an important feature required for good media search and retrieval. As CMML includes keyframes into clip tags through a URI reference, it is trivial to include them also into a query results page in HTML format.

In summary, the implemented extensions to existing Web applications show that our proposed approach provides a simple and effective means to extend the Web to time-continuous Web resources and make them first-class citizens. Experimental content sites prove that searching and browsing are enabled for both audio and video content. Initial integration with existing Web infrastructure has been achieved by extending an existing Web server and an existing Web search engine. A better integration of Annodex with existing Web browsers through browser plugins is the next logical step to improve integration.

9 Related standards

One may expect that the many existing standardization efforts in multimedia would already allow for the creation of a Continuous Media Web as we are proposing. However, this is not the case. The three most commonly pointed out

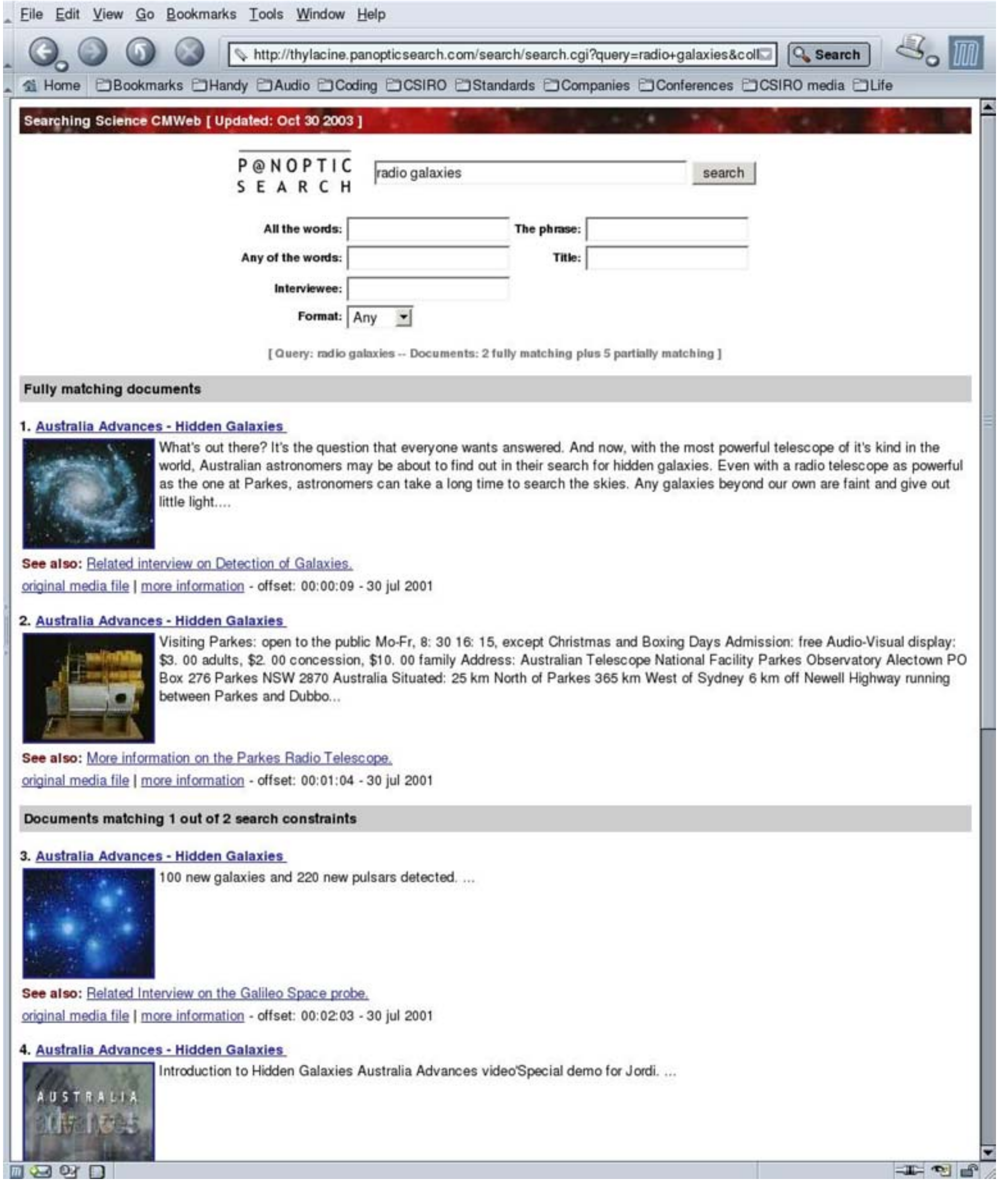

Fig. 6 Screenshot of a CMWeb search engine interface

standards that relate to our work are SMIL, MPEG-7, and MPEG-21.

9.1 SMIL

According to `http://www.w3.org/TR/smil20/`, the W3C's "Synchronized Multimedia Interaction Language," SMIL 2.0 [31], has the following two design goals:

- Define an XML-based language that allows authors to write *interactive multimedia presentations*. Using SMIL 2.0, an author can describe the temporal behavior of a multimedia presentation, associate hyperlinks with media objects, and describe the layout of the presentation on a screen.
- Allow reuse of SMIL syntax and semantics in other XML-based languages, in particular those that need to represent timing and synchronization. For example, SMIL 2.0 components are used for integrating timing into XHTML [31] and into SVG (Scalable Vector Graphics).

SMIL focuses on the authoring and presentation of interactive multimedia presentations composed of multiple media formats, encompassing animations, audio, video, images, streaming text, and text. A SMIL document describes the sequence of media documents to play back, including conditional playback, loops, and automatically activated hyperlinks. SMIL has outgoing hyperlinks and elements that can be addressed inside it using XPath [35] and XPointer [32].

Features of SMIL cover the following modules:

1. *Animation* provides for incorporating animations on a timeline.
2. *Content control* provides for runtime content choices and prefetch delivery.
3. *Layout* allows positioning of media elements on the visual rendering surface and control of audio volume.
4. *Linking* allows navigations through the SMIL presentation that can be triggered by user interaction or other triggering events. SMIL 2.0 provides only for inline link elements.
5. *Media objects* describes media objects that come in the form of hyperlinks to animations, audio, video, images, streaming text, or text. Restrictions of continuous media objects to temporal subparts (clippings) are possible, and short and long descriptions may be attached to a media object.
6. *Metainformation* allows description of SMIL documents and attachment of RDF metadata to any part of the SMIL document.
7. *Structure* structures a SMIL document into a head and a body part, where the head part contains information that is not related to the temporal behavior of the presentation and the body tag acts as a root for the timing tree.
8. *Timing and synchronization* provides for different choreographing of multimedia content through timing and synchronization commands.
9. *Time manipulation* allows manipulation of the time behavior of a presentation, such as control of the speed or rate of time for an element.
10. *Transitions* provides for transitions such as fades and wipes.
11. *Scalability* provides for the definition of profiles of SMIL modules (1–10) that meet the needs of a specific class of client devices.

So, with all these capabilities, does SMIL allow for the creation of webs of time-continuous resources?

SMIL is defined for creating interactive multimedia presentations – thus a SMIL document does not in the strict sense defined above represent a time-continuous Web resource as it does not actually contain time-continuous data – it is just an XML document. A combination of the SMIL XML file plus its externally referenced files of multiple media types also does not create a time-continuous Web resource as these files all have different timelines that they run toward. Every use of a SMIL file can result in a different experience and therefore a SMIL document is not a single, temporally addressable time-continuous data stream that compares directly to the Annodex format. Consequences of this issue on the possibilities of retrieval of a consistent presentation fragment of interest have been discussed in [6].

So, while SMIL does not compare to the Annodex format, could it be used to do the work of CMML?

SMIL is an XML-based authoring format for multimedia presentations. In that respect, it is far more advanced than CMML as it has many features to allow for diverse media to be composed together. The stream tag in CMML provides a very basic means of authoring multitrack media streams, but that's where the presentation authoring capabilities of CMML end. However, this is also a strength of CMML: it is not restricted to authoring presentations with audio or video, but it extends to any time-continuous data type, such as the above-mentioned acidity measurements, while SMIL is restricted to animations, audio, video, and streaming text as time-continuous data types.

In addition, CMML is designed to be serialized and interleaved into a binary bitstream format, where clips of time retain their metainformation together with their media content. SMIL, on the other hand, is a general XML file with a hierarchical structure that cannot generally be serialized. Even if it were possible to define an Annodex-specific subset of SMIL to support the authoring and markup creation for an Annodex resource, the serialization would be highly complex and infeasible for on-the-fly compositions. An example of such a SMIL file that contains RDF and externally referenced metadata, a decomposition into several clips (termed *clippings* in SMIL), and some anchors to the clips is given in Fig. 7.

The final point to look at with respect to SMIL is the hyperlinking functionality. As SMIL is an XML-based markup language, XPointer [32] and XPath [35] provide all the hyperlinking functionality to link to specific named tags or named time points. There is no general mechanism to address arbitrary time offsets through URI hyperlinks, though. As those do not actually exist since a SMIL presentation's timeline is defined by a user's interaction, this would not make sense. What is more important than the XML side of hyperlinking is, however, the hyperlinking into the time-continuous data themselves. As such a consistent binary file format does not exist within the SMIL-based technology, hyperlinking between time-continuous Web resources is not supported by SMIL. The simple extraction of subparts of media based on the specification of a temporal interval is a multistage complex process with SMIL that discourages its use for dynamic media fragment reuse.

SMIL 2.0 is a very powerful standard for creating multimedia presentations. It is not a standard for enabling searchable and crawlable webs of time-continuous resources. It integrates with the CMWeb where a recording of a single interactive session of a user with a SMIL presentation would be used as the basis for annotating and indexing. Such a recording would result in a time-continuous data stream that could be referenced in a CMML stream tag and could thus be annodexed and integrated into the CMWeb. If the recording is done intelligently, existing SMIL metainformation and hyperlinks can find their way into Annodex and thus create a rich resource.

9.2 MPEG-21

The ISO/MPEG MPEG-21 [5] standard is building an open framework for multimedia delivery and consumption. It thus focuses on addressing how to generically describe a set of content documents (called a "digital item") that belong together from a semantic point of view, including all the

```
<smil>
 <head>
  <meta id="m1" name="Producer" content="CSIRO" />
  <metadata id="meta-rdf">

  <!-- RDF and DC and smil metadata -->
  <rdf:RDF
   xmlns:rdf =
   "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:dc =
   "http://purl.org/metadata/dublin_core#"
   xmlns:smilmetadata =
   "http://.../smil-ns#" >

   <rdf:Description about="#intro"
    dc:Title="Introduction to CSIRO"
    dc:Description="
       Welcome to the CSIRO, the Commonwelath
       Scientific and Industrial Research
       Organisation of Australia."
    smilmetadata:Duration="33 secs">
   </rdf:Description>

   <rdf:Description about="#astronomy"
    dc:Title="Tidbinbilla Honeysuckle
              space tracking station"
    dc:Description="
       Our astronomers track the Galileo space
       probing station in Tidbinbilla."
    smilmetadata:Duration="42 secs">
   </rdf:Description>
  </rdf:RDF>
 </head>

 <body>
  <seq>
   <a href="http://www.csiro.au/">
    <video id="intro" src="index.mpg"
     clip-begin="0s"
     clipBegin="marker=#intro"/>
   </a>

   <a href="00-4-5.mpg">
    <video id="astronomy" src="index.mpg"
     clip-begin="33s"
     clipBegin="marker=#astronomy"
     alt="Tidbingilla Honeysucke
          space tracking station"
     longdesc="radio.html"/>
   </a>
  </seq>
 </body>
</smil>
```

Fig. 7 SMIL file that is analogous to a CMML file

information necessary to provide services on these digital items.

As an example, consider a music CD album. When it is turned into a “digital item,” the album is described in an XML document that contains references to the cover image, the text on the CD cover, the text on an accompanying brochure, references to a set of audio files that contain the songs on the CD, ratings of the album, rights associated with the album, information on the different encoding formats in which the music can be retrieved, different bitrates that can be supported when downloading, etc. This description supports everything that you would want to do with a digital CD album: it allows you to manage it as an entity, describe it with metadata, exchange it with others, and collect it as an entity.

An MPEG-21 document thus does not typically describe just one media document, but rather several and additionally describes other types of content such as images or graphs. An MPEG-21 document is an XML document that describes how documents in a set relate to each other, how they should be distributed and managed.

As with SMIL, an MPEG-21 document is an XML document with hyperlinks to the content it describes. Therefore, an MPEG-21 document is also not a time-continuous document as defined above, excluding it from enabling deep hyperlinked, searchable, and browsable webs of time-continuous resources in a nonstaged fashion.

In contrast to being interested in the deep structure of single time-continuous Web resources as is the aim of Annodex, MPEG-21 addresses how collections of documents are handled. It can therefore be used to describe collections of Web resources that belong together for a semantic reason, such as the different songs, the cover image, and related metainformation of the CD album described above. MPEG-21 can therefore be used to describe, for example, a Web of Annodex, JPEG, and HTML resources that represent the CD.

Interestingly, MPEG-21 has come up against the problem of how to address content inside an MPEG-21 digital item through hyperlinks. This is required as metainformation given in an MPEG-21 file can, for example, relate to a segment of an MPEG-21 content file. Here, Annodex resources and the proposed temporal hyperlinks provide a solution for temporal addressing of time-continuous data. As MPEG-21 requires a more general solution to allow, for example, addressing of spatial regions or spatiotemporally moving regions, a different addressing scheme is being discussed within MPEG-21 that will require a two-stage process where first the resource is resolved and then the fragment addressing is performed depending on the capabilities of the addressed resource.

The aims of MPEG-21 are orthogonal to the aims that we pursue. MPEG-21 does not allow the creation of a web of time-continuous resources; however, it does provide for the distribution, management, and adaptation of sets of Web resources that are related to each other. It covers the bigger picture in which time-continuous Web resources play an equal part to all other resources, but it does not allow them to become equal. The Annodex format is therefore a prime candidate to become part of the formats that MPEG-21 digital items can hold to allow time-continuous Web resources to be handled on equal terms.

9.3 MPEG-7

The ISO/MPEG MPEG-7 [15] standard is an open framework for describing multimedia content. It provides a large set of description schemes to create markup in XML format. MPEG-7's markup is not restricted to textual information only – in fact it is tailored to allow for the description of audiovisual content with low-level image and audio features as extracted through signal processing methods. It also has basically no resemblance to HTML as it was *not* built with a particular focus on Web applications.

An MPEG-7 document is an XML file that contains any sort of metainformation related to a media document. The MPEG-7 committee has developed a large collection of description schemes that cover the following aspects:

1. *Content management*: Creation and production metainformation, media coding, storage and file format information, and content usage information.
2. *Content description*: Description of structure and semantics of media content. The structural tools describe the structure of the content in terms of video segments, frames, still and moving regions, and audio segments. The semantic tools describe the objects, events, and notions from the real world that are captured by the AV content.
3. *Content navigation and access*: Descriptions that facilitate browsing and retrieval of audiovisual content by defining summaries, partitions and decompositions, and variations of the audiovisual material.
4. *Content organization*: Descriptions for organizing and modeling collections of audiovisual content and of descriptions.
5. *User interaction*: Descriptions of user preferences and usage history pertaining to the consumption of the media material.

As with SMIL and MPEG-21, an MPEG-7 document is an XML document with references to the content it describes. Hyperlinking into time-continuous resources with MPEG-7 is a two-stage process where in a first stage linking to descriptions of the requested fragment of an MPEG-7 description of a media document is performed and from there, with the specification of a temporal offset, it is possible to perform the file offset function on the content. Obviously, an MPEG-7 document is also not a time-continuous document as defined above, excluding it from enabling deep hyperlinked, searchable, and browsable webs of time-continuous resources in a nonstaged fashion.

A representation of the content of a CMML file in the way MPEG-7 would represent it is shown in Figs. 8 and 9.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001"
 xmlns:xsi=
  "http://www.w3.org/2001/XMLSchema-instance"
 xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001"
 xmlns:semantics=
  "urn:mpeg:mpeg7:cs:SemanticRelationCS:2001">

 <Description xsi:type="ContentEntityType">
  <MultimediaContent xsi:type="VideoType">
   <Video>

    <MediaLocator>
     <MediaUri>index.mpg</MediaUri>
    </MediaLocator>

    <MediaTime>
     <MediaTimePoint>T00:00:00</MediaTimePoint>
     <MediaDuration>PT5M00S</MediaDuration>
    </MediaTime>

    <Classification>
     <Subject>
      <FreeTextAnnotation>
       The Research Hunter
      </FreeTextAnnotation>
     </Subject>
     <Language type="original">en</Language>
    </Classification>
```

Fig. 8 MPEG-7 file that is analogous to a CMML file (Part 1)

A more extensive example description of a similar markup was published by the *Harmony Project* (see [9]).

While MPEG-7 can be used to create markup similar to the one possible in CMML, it is a difficult task to identify the required tags from the several thousands available through MPEG-7. It would be necessary to define a profile of MPEG-7. Specifications in this profile would need to be serialized and interleaved into a binary bitstream format to achieve searchable and surfable time-continuous Web resources such as Annodex. As is the case with SMIL, MPEG-7 is not designed for this serialization but is a general XML file with a hierarchical structure. Therefore, the serialization can be expected to be complex and infeasible for on-the-fly compositions.

MPEG-7 is a powerful collection of description tools for multimedia content covering all sorts of metainformation necessary with media content with no particular view on networking-related issues and scalability over the Web. However, MPEG-7 does not enable the creation of webs of media, nor of webs of time-continuous data. MPEG-7 has a rich set of description schemes, and the creation of metaschemes for CMML will require the specification of such collections of sensible metatags. A direct integration of MPEG-7 markup with the CMWeb may be through referencing of richly marked-up MPEG-7 files from inside an Annodex format bitstream. Also, Annodex files can be used as the basis for metainformation created in the MPEG-7 format.

```
      <TemporalDecomposition
       gap="false" overlap="false">

       <VideoSegment id="intro">
        <MediaTime>
         <MediaTimePoint>T00:00:00</MediaTimePoint>
         <MediaDuration>PT00M33S</MediaDuration>
        </MediaTime>
        <TextAnnotation>
         <FreeTextAnnotation>
          Welcome to CSIRO, the Commonwealth
          Scientific and Industrial Research
          Organisation of Australia.
         </FreeTextAnnotation>
        </TextAnnotation>
        <Relation
         type="semantics:key"
         target="/media_images/index1.jpg"/>
        <Relation
         type="semantics:relation"
         target="http://www.csiro.au/"/>
       </VideoSegment>

       <VideoSegment id="astronomy">
        <MediaTime>
         <MediaTimePoint>T00:00:33</MediaTimePoint>
         <MediaDuration>PT00M42S</MediaDuration>
        </MediaTime>
        <TextAnnotation>
         <FreeTextAnnotation>
          Our astronomers track the Galileo space
          probing station in Tidbinbilla.
         </FreeTextAnnotation>
        </TextAnnotation>
        <Relation
         type="semantics:key"
         target="/media_image/index2.jpg"/>
        <Relation
         type="semantics:relation"
         target="00-5-4.mp3"/>
       </VideoSegment>

       <VideoSegment ...
       </Video Segment>

      </TemporalDecomposition>

     </Video>
    </MultimediaContent>
   </Description>
  </Mpeg7>
```

Fig. 9 MPEG-7 file that is analogous to a CMML file (Part 2)

10 Summary and outlook

This article presented the Annodex technology, which extends the familiar searching and surfing capabilities of the World Wide Web to time-continuous data. This implies that media data are not handled as atomic at the file level, but conceptually broken up into clips and thus made deeply addressable. At the core of the technology are the Continuous Media Markup Language CMML, the Annodex interchange format, and clip- and time-referencing URI hyperlinks. These enable the extensions of the Web to a Continuous Media Web with Annodex browsers, Annodex servers, Annodex content, and Annodex search engines. These applications together extend the Web to become a powerful distributed information retrieval system for multiple media types where different media types are handled equally.

Each one of the discussed existing multimedia standards addresses a related but distinct issue related to our work. Combined, all these standards, including the CMWeb, form a solid basis for a more integrated and interoperable use of distributed media on the Internet. SMIL allows authoring of highly interactive multimedia content, MPEG-7 allows for the creation of rich XML annotations for media content, Annodex allows for the exchange of enriched clips of media content in a standardized format and also allows search engines to crawl and index time-continuous data, and MPEG-21 allows for the management of collections of documents of multiple media types.

Another interesting standard to look at in this context is the Semantic Web (`http://www.w3.org/2001/sw/`) with RDF (the Resource Description Framework [34]) at its core. RDF is a general-purpose language for representing information on the Web. It is based on XML and URIs to attach information to content. In this context, Annodex breaks up files or streams of time-continuous data into smaller segments, enabling RDF to attach information to clips of content. This relationship must be explored further in future work.

Having published stable specifications of CMML, Annodex, and the temporal URI formats, we are now building plugins for common Web browsers and researching search algorithms that better exploit the Annodex and CMML formats and provide for high-quality search results in media information retrieval. The combination with other types of Web resources and the presentation of search results are particular facets of this research. We are also investigating how to exploit the features of Annodex to create more-powerful user interfaces for multimedia applications. These can in particular support interactive information exploration tasks and improve the CMWeb browsing experience. Intensive automated audiovisual content analysis research in recent years has created a solid foundation to explore new ways of automating CMML authoring, which is another research topic that we are interested in and that will support the uptake of Annodex.

Acknowledgements The authors greatly acknowledge the comments, contributions, and proofreading of Claudia Schremmer, who is making use of the Continuous Media Web technology in her research on metainformation extraction of meeting recordings. The authors are also grateful to the reviewers for their suggestions for improvements, which have contributed to improve the article.

References

1. Annodex.net: Open Standards for Annotating and Indexing Networked Media. `http://www.annodex.net`. [Accessed Nov. 2004]

2. Baeza-Yates, R., Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval, 1st edn. Addison-Wesley, Boston (1999)
3. Berners-Lee, T., Fielding, R., Irvine, U., Masinter, L.: Uniform Resource Identifiers (URI): Generic Syntax. http://www.ietf.org/rfc/rfc2396.txt (1998) [Accessed Nov. 2004]
4. Berners-Lee, T., Fischetti, M., Dertouzos, M.: Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by its Inventor. Harper, San Francisco (1999)
5. Burnett, I., de Walle, R.V., Hill, K., Bormans, J., Pereira, F.: MPEG-21: Goals and Achievements. IEEE Multimedia J. **10**(4), 60–70 (2003)
6. Celentano, A., Gaggi, O.: Querying and browsing multimedia presentations. In: Tucci, M. (ed.) MDIC 2001, 2nd International Workshop on Multimedia Databases and Image Communication. Lecture notes in computer science, vol. 2184, pp. 105–116. Springer, Berlin Heidelberg New York (2001)
7. Dublin Core Metadata Initiative: Dublin Core Metadata Element Set, Version 1.1. http://dublincore.org/documents/2003/02/04/dces (2003) [Accessed Nov. 2004]
8. Fielding, R., Gettys, J., Mogul, J., Nielsen, H., Masinter, L., Leach, P., Berners-Lee, T.: Hypertext Transfer Protocol–HTTP/1.1.http://www.ietf.org/rfc/rfc2616.txt (June 1999) [Accessed Nov. 2004]
9. Hunter, J.: An application profile which combines DC and MPEG-7 for simple video description. ViDE Video Access Group. http://www.metadata.net/harmony/video_appln_ profile.html [Accessed Feb. 2002]
10. International Press Telecommunications Council: Specification NewsML 1.2. http://www.newsml.org/pages/spec main.php (2003) [Accessed Nov. 2004]
11. Jain, R., Hampapur, A.: Metadata in video databases. SIGMOD Rec. **23**(4), 27–33 (1994)
12. Jansen, B., Goodrum, A., Spink, A.: Searching for multimedia: video, audio, and image Web queries. World Wide Web J. **3**(4), 249–254 (2000)
13. Laurie, B., Laurie, P.: Apache: the Definitive Guide, 1st edn. O'Reilly, Sebastopol, CA (1999)
14. Little, T., Ghafoor, A.: Interval-based conceptual models for time-dependent multimedia data. Knowl. Data Eng. **5**(4), 551–563 (1993)
15. Martinez, J., Koenen, R., Pereira, F.: MPEG-7: the Generic Multimedia Content Description Standard. IEEE Multimedia J. **9**(2), 78–87 (2002)
16. Mulhem, P., Martin, H.: From database to Web multimedia documents. Multimedia Tools Appl. J. **20**(3), 263–282 (2003)
17. Naphade, M.R., Huang, T.S.: A probabilistic framework for semantic indexing and retrieval in video. In: IEEE International Conference on Multimedia and Expo (I), pp. 475–478 (2000)
18. NCSA HTTPd Development Team. The Common Gateway Interface (CGI). http://hoohoo.ncsa.uiuc.edu/cgi/ (1995) [Accessed Nov. 2004]
19. Nilsson, M.: ID3 tag version 2.4.0 – Main Structure. http://www.id3.org/id3v2.4.0-structure.txt (2000) [Accessed Nov. 2004]
20. Pfeiffer, S.: The Ogg Encapsulation Format Version 0. http://www.ietf.org/rfc/rfc3533.txt (2003)
21. Pfeiffer, S., Parker, C., Pang, A.: Specifying time intervals in URI queries and fragments of time-based Web resources (work in progress). http://www.ietf.org/internet-drafts/draft-pfeiffer-temporal-fragments-03.txt (2004) [Accessed Dec. 2004]
22. Pfeiffer, S., Parker, C., Pang, A.: The Annodex annotation and indexing format for time-continuous data files, version 3.0 (work in progress). http://www.ietf.org/internet-drafts/draft-pfeiffer-annodex-02.txt (2004) [Accessed Dec. 2004]
23. Pfeiffer, S., Parker, C., Pang, A.: The Continuous Media Markup Language (CMML), version 3.0 (work in progress). http://www.ietf.org/internet-drafts/draft-pfeiffer-cmml-02.txt (2004)
24. Pfeiffer, S., Parker, C., Schremmer, C.: Annodex: a simple architecture to enable hyperlinking, search and retrieval of time-continuous data on the Web. In: Proceedings of the IEEE 5th ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR), pp. 87–93. Berkeley, CA (2003)
25. Rutledge, L., Schmitz, P.: Improving media fragment integration in emerging Web formats. In: Proceedings of the International Conference on Multimedia Modeling 2001 (MMM01), pp. 147–166. CWI, Amsterdam (2001)
26. Schulzrinne, H., Rao, A., Lanphier, R.: Real Time Streaming Protocol (RTSP). http://www.ietf.org/rfc/rfc2326.txt (1998) [Accessed Nov. 2004]
27. Smith, J.R., Chang, S.-F.: Image and video search engine for the World Wide Web. In: Storage and Retrieval for Image and Video Databases (SPIE), pp. 84–95 (1997)
28. Sullivan, D.: Multimedia Search Engines: Image, Audio & Video Searching. http://searchenginewatch.com/links/article.php/2156251 (2003) [Accessed Nov. 2004]
29. World Wide Web Consortium (W3C): HTML 4.01 Specification. http://www.w3.org/TR/html4/ (1999) [Accessed Nov. 2004]
30. World Wide Web Consortium (W3C): XSL Transforms (XSLT) Version 1.0. http://www.w3.org/TR/xslt/ (1999) [Accessed Nov. 2004]
31. World Wide Web Consortium (W3C): Synchronized Multimedia Integration Language (SMIL 2.0). http://www.w3.org/TR/smil20/ (2001) [Accessed Nov. 2004]
32. World Wide Web Consortium (W3C): XML Pointer Language (XPointer). http://www.w3.org/TR/xptr/ (August 2002) [Accessed Nov. 2004]
33. World Wide Web Consortium (W3C): Extensible Markup Language (XML) 1.1. http://www.w3.org/TR/xml11/ (2004) [Accessed Nov. 2004]
34. World Wide Web Consortium (W3C): RDF/XML Syntax Specification. http://www.w3.org/TR/rdf-syntax-grammar/ (2004) [Accessed Nov. 2004]
35. World Wide Web Consortium (W3C): XML Path Language (XPath) 2.0. http://www.w3.org/TR/xpath20/ (2004) [Accessed Nov. 2004]
36. Building a New Era of Open Multimedia: http://www.xiph.org [Accessed Nov. 2004]
37. Young, S., Foote, J., Jones, G., Sparck, K., Brown, M.: Acoustic indexing for multimedia retrieval and browsing. In: Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '97), April 1997, vol. 1, pp. 199–202. Munich, Germany (1997)