Audio Meta Data Transcription from Meeting Transcripts for the Continuous Media Web

Claudia Schremmer[1], Steve Cassidy[2], and Silvia Pfeiffer[1]

[1] *CSIRO ICT–Centre, Locked Bag 17, North Ryde, NSW 1670, Australia*

[2] *Macquarie University, Department of Computing, North Ryde, NSW 2109, Australia*

Correspondence should be addressed to Claudia Schremmer (`claudia.schremmer@csiro.au`)

ABSTRACT

The Continuous Media Web (CMWeb) integrates time–continuous media into the searching, linking, and browsing functionality of the World Wide Web. The file format underlying the CMWeb technology, *Annodex*, streams the media content multiplexed with XML–markup in the Continuous Media Markup Language (*CMML*). CMML contains information relevant to the whole media file (e.g., title, author, language) as well as time–sensitive information (e.g., topics, speakers, time–sensitive hyperlinks). This paper discusses the challenges of automatically generating Annodex streams from complex annotated recordings collected for use in linguistic research. We are particularly interested in annotated recordings of meetings and teleconferences and regard Annodex and its media browsing paradigm as a novel and rich way of interacting with such recordings. The paper presents our experiments with generating CMML and their corresponding Annodex files from hand annotated meeting recordings.

1. INTRODUCTION

In mid–2003, CSIRO launched the Continuous Media Web (CMWeb) [1, 13, 12], designed to solve the problem of "dark matter" of time–continuous media such as audio and video on the Web. The motivation for the proposed streamable annotated and indexed Annodex file format is the integration of time–continuous media into the URL–based hyperlinking paradigm, resulting in "surfable" and "searchable" media. In this new proposed standard which has been submitted to the IETF for standardization [10], users can not only hyperlink to, e.g., an audio file, but *into* a specific time interval containing the information sought–after [9]. The audio itself is annotated with HTML–like markup information and might link to yet other resources on the Web (e.g., text, audio, images, video), enabling the user to switch from one digital resource to another, just like "browsing" a Web site. The technology is based on the observation that the data stream of existing time–continuous data can be subdivided into segments based on a semantic concept such as topic boundaries, and that this structure enables access to interesting subparts or *clips* of the stream.

One significant source of annotated time–continuous media is that of linguistic corpora. These are collections of language data, often speech, audio and video, which have been annotated for their linguistic content. The annotations range from simple time–aligned transcriptions to complex structures of encoded dialogue including forward and backward relations amongst turns, annotations of overlapping speech etc. The speech recognition community is now targeting meeting recordings. The term "meeting" is broad, embracing face–to–face meetings as well as teleconferences. They range from formal presentations over organized group meetings to informal "brain storming" sessions. In all cases, the nature of the recording is that the number of participants is not restricted (and often not known) and the language used is informal and spontaneous.

This paper describes some experiments in transforming meeting transcriptions into the streaming media format Annodex of the Continuous Media Web. This is motivated in part by the need to automate the creation of Annodex content: Speech is a rich source of content for media data. The experiments also form an additional part of the validation of the CMWeb standards for streaming and linking time–continuous media.

Section 2 introduces the Continuous Media Markup Language (*CMML*) that allows authoring of textual markup for time–continuous data in Annodex format. Section 3 outlines the concept of linguistic annotations for meeting recordings. Section 4 presents the core of this paper and describes the means of transcribing complex linguistic transcripts into CMML. In Section 5, we summarize the benefits from our approach. Some future challenges are discussed in Section 6.

2. CONTINUOUS MEDIA MARKUP LANGUAGE

The Annodex file format [1, 10] for the Continuous Media Web contains a media data stream interspersed with islands of XML markup that contains annotations and supports hyperlinking to other clips and Web resources. Web server extensions allow search engines to retrieve the XML markup representing the Annodex file. For producing an Annodex file, a markup file written in the Continuous Media Markup Language CMML [11] is interleaved with the media stream. Figure 1 illustrates the merging process. Here, *anxenc* is short for *Annodex encode*.

2.1. CMML Syntax

A sample CMML file is given below. The XML markup of a `head` tag embraces information for the complete media document. It contains structured textual annotations in `meta` tags as well as unstructured textual annotations in the `title` tag. Structured annotations are `name-value` pairs which may follow a new or an existing meta data annotation scheme (such as the Dublin Core [6] used in the example below). Following the `head` element are one or more `clips`, i.e., semantic units of the media file, which might contain meta data, hyperlinks,

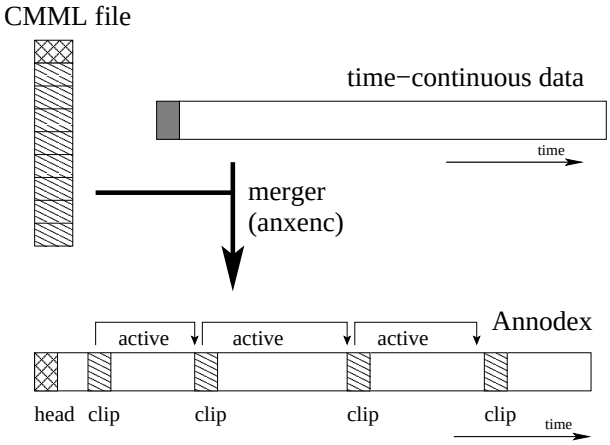

Fig. 1: Merging XML markup in CMML format with a media file.

keyframe images, and textual descriptions for a temporal segment of the media document. The `clips` are to be interleaved time–synchronously into the Annodex bitstream.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE cmml SYSTEM "cmml.dtd">
<cmml>
 <stream timebase="npt:0">
   <import mimetype="audio/wav"
    src="projectmeeting3.wav"
    start="npt:0"/>
 </stream>
 <head>
   <title>
     Project Meeting Nov 2003
   </title>
   <meta name="DC.author" content="CSIRO"/>
   <meta name="DC.date.created"
    content="2003/11/10"/>
   <meta name="DC.description"
    content="Meeting Recording"/>
   <meta name="DC.format" content="audio"/>
   <meta name="DC.language" content="en"/>
   [...]
 </head>
 <clip id="agenda" start="npt:0">
   <a href="http://foo.bar/agenda.htm">
     Agenda of the meeting on 2003/11/10
   </a>
   <img src="http://foo.bar/chair.jpg"/>
   <desc>
```

```
    Peter outlines the agenda for the meeting.
    It includes...
  </desc>
 </clip>
 <clip id="topic1" start="npt:230" end="npt:423">
  <a href="http://foo.bar/topic1.htm">
    Hyperlink to the HTML pages of the
    meeting slides, topic 1.
  </a>
  <desc>
    Topic 1: How can we make a meeting more
    efficient? There are several options...
  </desc>
 </clip>
 <clip id="topic2" start="npt:430">
  [...]
 </clip>
</cmml>
```

As the sample file shows, the structure of the CMML annotations is fairly simple. A time–continuous media file is subdivided into clips defined by a start time and an end time. The CMML DTD (http://www.annodex.net/DTD/cmml.dtd) allows most of the tags and parameters (such as the `end` parameter in the `clip` tag) to be optional. Each clip might contain amongst others a hyperlink `<a>` to a related Web resource, an image `<img>` representing the content of the clip, a free-text description `<desc>` of its content, and a set of meta data `<meta>` providing structured annotations for the clip.

The user experience of consuming an Annodex media file is such that while, e.g., listening to an audio file, the annotations and links change over time: During the playback of a recorded speech file, the islands of CMML markup data that are defined in the `clip` tags and that are interspersed with the media file itself, enter and quit their active periods through their `start` and optional `end` tags. If at any arbitrary offset into the speech file, there is a clip active for that period, its markup is extracted. If the clip under consideration happens to have a hyperlink defined through its `<a>` tag, the user may decide to activate it during the duration of this clip.

2.2. Multitrack Paradigm

So far, we introduced the basic structure of CMML. For the break–down of linguistic transcripts (see Section 3) into the rather linear consumption structure of CMML for Annodex, we make use of CMML's multitrack paradigm, which enables a set of temporally overlapping annotation tracks per Annodex file. This is implemented through the definition of a `track` attribute within a `clip` tag. If no `track` is set, it defaults to `track="default"`. See Figure 2 for how a time–continuous media file is interspersed with islands of CMML annotations that belong to different annotation tracks. Whereas clips within the same annotation track cannot overlap, the active period for the markup of clips belonging to different tracks might overlap in time.

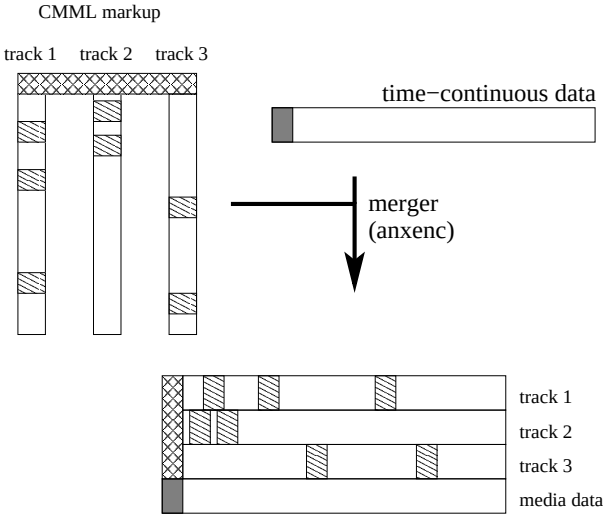

Fig. 2: The multitrack paradigm of CMML: A media file can be annotated with an arbitrary number of annotation tracks. The resulting Annodex bitstream may contain more than one active clip for each given point in time.

The multitrack paradigm allows the consumer of a meeting recording to change his viewing angle. If, for example, there are three different topic tracks to a recording, the user can choose which topic he is interested in and only use the annotations related to his choice. For our speech annotations, we have experimented with different tracks according to the speaker. If, for example, we have a speaker `peter`, the `clip` syntax as described in Section 2.1 is enhanced by the `track="peter"` attribute:

```
<clip id="id25" track="peter"
      start="102.300000" end="104.310000">
```

The realization of how this is implemented in the user interface is left to the application, i.e., to the respective Annodex browser. The generic implementation allows the user to select his annotation track of interest from a list. A specific implementation for, e.g., multispeaker annotation tracks could also represent all the speakers and their speech in different sections of the screen in parallel.

3. LINGUISTIC ANNOTATION

Linguistic data is annotated in many ways and includes complex interlinking between elements. In particular, annotations can often involve multiple *overlapping* hierarchies: For example describing two simultaneous speakers, two concurrent topics for one piece of spoken language, or alternate structural views of a piece of language data. Hence, the underlying data structure of linguistic annotations is highly complex. One of the goals of recent work in the area of linguistic annotations has been to define a standard data model able to represent all the various kinds of annotation: From simple time–aligned transcriptions to complex structures including forward and backward relations amongst turns, annotations of overlapping speech etc. One such proposal is the Annotation Graph [2] which represents annotations as a directed acyclic graph. Some of our recent work in this area has been trying to identify a sufficiently powerful query language for use on annotations [3, 4].

For this work, we have used linguistic transcriptions in the Universal Transcription Format UTF [15]. The goal of this linguistic annotation format is to capture the orthography of spoken words in recordings of speech and can also include annotations which associate certain signal, speaker, and content conditions with the speech and its transcription. The convention is formatted using SGML markup. SGML tags identify the location of the speech within the recorded waveform using start/end time attributes, and they identify the corresponding transcription by enclosing it within the span of these tags. UTF has at its core a notion of speaker turns. A speaker turn marks the extent and content of the words spoken by a single speaker, spoken in a particular speaking style [15].

There are four categories of tags in a UTF document: structural, state change, pseudo bracketing, and lexical. The pseudo bracketing tags do not use the SGML hierarchy of begin and end tags, but instead explicit begin tags and end tags (e.g. `<b_unclear>unclear speech<e_unclear>`).

In the following, we use hand annotated meeting transcripts in UTF format developed in the ICSI Meeting Room Project [7] that were used in the NIST Rich Transcription Evaluation Spring 2004 [8]. We will use the following excerpt of the ICSI meeting transcription `ICSI_20010208-1430.utf` to monitor its changes throughout our transcoding process during the remain of this paper.

```
<utf dtd_version="utf-1.0" audio_filename="ICSI_
 20010208-1430" scribe="dummy" language="English"
 version="20" version_date="08-Aug-2001">
<conversation_trans recording_date="unknown">

[...]
<turn startTime="101.008000" endTime="102.383000"
 speaker="me011" spkrType="male" channel="5">
 but that <contraction e_form="[wo=>will][n't=>
 not]">won't take too long .
</turn>
<turn startTime="101.040000" endTime="101.510000"
 speaker="me013" spkrType="male" channel="2">
 Right .
</turn>
<turn startTime="102.300000" endTime="104.310000"
 speaker="me013" spkrType="male" channel="2">
 <acronym>O<acronym>K , agenda items ,
</turn>
<turn startTime="104.750000" endTime="106.241000"
 speaker="me013" spkrType="male" channel="2">
 <nonlexeme>Uh , we have digits ,
</turn>
[...]

</conversation_trans>
</utf>
```

4. TRANSCRIPTION INTO CMML

The scope envisaged in this work is to define structures within linguistic annotations that are hierarchical, thus allowing XML concepts and tools to manipulate linguistic data. This work has now lead to an investigation of the use of XSLT stylesheets

[5] for transforming (hand) annotated meeting transcriptions into display formats like CMML.

This section describes our work to automatically generate a CMML file for the Annodex file format from annotated meeting transcripts in the UTF format. The advantage of using these speech transcripts is that there is a large number available. Hence, it allows us to concentrate on the high–level problems of how to re–purpose existing material for our specific needs.

4.1. XML–ification of UTF

As we have seen in Section 3, linguistic annotations in the UTF format are not strictly XML. In order to be able to engage XML tools to retrieve the interesting information, we modified the original transcripts `ICSI_date-time.utf` as follows:

- `<b_noscore ATT>` has the corresponding closing tag `<e_noscore>`. This was modified to `<b_noscore ATT />`.
- `<b_unclear>unclear speech<e_unclear>` was modified to `<b_unclear>unclear speech</b_unclear>`.
- `<contraction ATT>` was modified to `<contraction ATT />`.
- `<acronym>O<acronym>K` was modified to `<acronym>O</acronym><acronym>K</acronym>`.
- `<nonspeech>swallow` was modified to `<nonspeech>swallow</nonspeech>`.
- ...and similar modifications...

On our sample file `ICSI_20010208-1430.utf`, this has the following effect:

```
<utf dtd_version="utf-1.0" audio_filename="ICSI_
 20010208-1430" scribe="dummy" language="English"
 version="20" version_date="08-Aug-2001">
<conversation_trans recording_date="unknown">

[...]
<turn startTime="101.008000" endTime="102.383000"
 speaker="me011" spkrType="male" channel="5">
 but that <contraction e_form="[wo=>will]
 [n't=>not]" />won't take too long .
</turn>
<turn startTime="101.040000" endTime="101.510000"
 speaker="me013" spkrType="male" channel="2">
 Right .
</turn>
<turn startTime="102.300000" endTime="104.310000"
 speaker="me013" spkrType="male" channel="2">
 <acronym>O</acronym><acronym>K</acronym> ,
 agenda items ,
</turn>
<turn startTime="104.750000" endTime="106.241000"
 speaker="me013" spkrType="male" channel="2">
 <nonlexeme>Uh</nonlexeme> , we have digits ,
</turn>
[...]

</conversation_trans>
</utf>
```

4.2. XSLT

We used an XSLT stylesheet [5] to transform the (modified) UTF files of hand annotated linguistic transcripts into CMML markup.

4.2.1. CMML's head Tag

Since both UTF and CMML markup contain much meta information, we were able to retrieve most of the meta information necessary in the CMML `head` tag directly from the UTF file. The XSLT code excerpt below produces the CMML `head` tag with its corresponding meta data that are valid for the complete media file.

```
<xsl:template match="utf">

<head>
<title>
<xsl:choose>
 <xsl:when test="conversation_trans">
  <xsl:text>Transcription of Meeting Recording:
  </xsl:text>
  <xsl:value-of select="@audio_filename" />
 </xsl:when>
 <xsl:otherwise>
  <xsl:text>The source file is not a Meeting
   Recording in utf format</xsl:text>
 </xsl:otherwise>
</xsl:choose>
</title>
<meta>
 <xsl:attribute name="name">
  <xsl:text>DC.DESCRIPTION</xsl:text>
 </xsl:attribute>
 <xsl:attribute name="content">
```

```
  <xsl:text>Automatically transformed linguistic
  transcription of Meeting recording </xsl:text>
  <xsl:value-of select="@audio_filename" />
  <xsl:text> into CMML by an xslt script.
  </xsl:text>
 </xsl:attribute>
</meta>
<meta>
 <xsl:attribute name="name">
  <xsl:text>DATE.TRANSCRIBED</xsl:text>
 </xsl:attribute>
 <xsl:attribute name="content">
  <xsl:value-of select="@version_date" />
 </xsl:attribute>
</meta>
[...]
</head>

</xsl:template>
```

The result of this script run on our sample file `ICSI_20010208-1430.utf` is the following CMML code

```
<head>
 <title>Transcription of Meeting Recording:
  ICSI_20010208-1430
 </title>
 <meta name="DC.DESCRIPTION" content=
  "Automatically [...] ICSI_20010208-1430
  into CMML by an xslt script."/>
 <meta name="DATE.TRANSCRIBED" content=
  "08-Aug-2001"/>
[...]
</head>
```

4.2.2. CMML's clip Tags

The following code excerpt transforms `turn` tags into `clip` tags with their respective attributes.

```
<xsl:template match="turn">
<clip>
  <xsl:attribute name="id">
    <xsl:value-of select="generate-id(.)" />
  </xsl:attribute>
  <xsl:attribute name="track">
    <xsl:value-of select="@speaker" />
  </xsl:attribute>
  <xsl:attribute name="start">
    <xsl:value-of select="@startTime" />
  </xsl:attribute>
  <xsl:attribute name="end">
    <xsl:value-of select="@endTime" />
  </xsl:attribute>
[...]
</clip>
</xsl:template>
```

In each clip, we are most interested in the content of the free–text description tags `<desc>[...]</desc>` since their content is searched and indexed by Web search engines. Due to the nature of the UTF source files, we did not look for key frame images for the `<img>` tags or for time–sensitive hyperlinks in the `<a>` tags although this is an interesting future research area.

The result of the above transformation is a subdivision into clips that span only a very short timeline — usually about 0.5 to 3 seconds duration, depending on the UTF source file. With the concept in CMML being that a clip is a semantic subdivision, we aimed at merging consecutive clips if they belong to the same track, i.e., speaker. When the following test on the speaker relative to the following speaker is true,

```
<xsl:when test="@speaker=substring(following-
     sibling::turn/@speaker,1)">
[...]
<\xsl:when>
```

the `clips` are merged. Hence, a series of consecutive `clips` with the same `track` attribute, like the last three clips in our standard excerpt:

```
<clip id="id26" track="me011"
 start="101.008000" end="102.383000">
 <desc>
  but that won't take too long .
 </desc>
</clip>
<clip id="id27" track="me013"
 start="101.040000" end="101.510000">
 <desc>
  Right .
 </desc>
</clip>
<clip id="id28" track="me013"
 start="102.300000" end="104.310000">
 <desc>
```

```
  OK , agenda items .
 </desc>
</clip>
<clip id="id29" track="me013"
 start="104.750000" end="106.241000">
 <desc>
  Uh , we have digits .
 </desc>
</clip>
```

result in the more compact clip presentation:

```
<clip id="id26" track="me011"
 start="101.008000" end="102.383000">
 <desc>
  but that won't take too long .
 </desc>
</clip>
<clip id="id27" track="me013"
 start="101.040000" end="106.241000">
 <desc>
  Right . OK , agenda items . Uh , we
  have digits .
 </desc>
</clip>
```

For this merging, not only two consecutive turns have to have the same `speaker` attribute, but a time threshold between the `endTime` tag of the previous and the `startTime` tag of the consecutive turn applies. This threshold was set to 2.5 seconds based on an indication that a break longer than this time period often creates a semantic break in the flow of the speech.

5. THE USER EXPERIENCE

What have we achieved so far? We started out with a time–continuous audio file of a meeting recording and its corresponding hand annotated file of linguistic transcriptions. Meetings are informal and spontaneous, and the meeting transcriptions reflect this. They contain many details of linguistic interest like:

- `Ther<fragment /> there`
- `Why don't you summarize the <!--PhrasalFragment-->`
- `We <mispronounced /><!--prolly-->probably [...]`
- `<nonspeech>swallow</nonspeech>`

Our goal within the Continuous Media Web is to establish a file format that is based on the time–continuous media plus meta information to cater for the needs of a Web with its searching and surfing paradigm. The focus is less on linguistic analysis of the recorded speech but rather to provide the user with a pleasant consuming experience of the underlying media file.

The stress in the results of the previous section shall be posed on the fact that a `clip` annotation as above contains a lot of information that we are interested in: `track, start, end` attributes. But most importantly, it also contains free-text annotation in the `<desc>` tags. Recalling that the CMWeb technology has been built to allow Web search engines to crawl and index this meta information, the Annodex file resulting from the merger of CMML and the speech file is fully searchable. In our example above, a (textual) search for "agenda items" would link exactly into our speech segment with `start="101.040000"` and `end="106.241000"`.

Hence, we have targeted the following challenges:

- Strip the linguistic annotation from those only of linguistic interest (by not processing them with the XSLT stylesheet).
- Make use of the track paradigm of CMML to break down the linear recording with its non–linear linguistic annotation into a meaningful semantic partition.
- Subdivide the speech recording into semantically meaningful clips.

6. FUTURE WORK

For this work, we have focused on subdividing a CMML file into multiple tracks according to the *speaker*. An interesting issue is to explore the CMML's multitrack paradigm for various *topics*. This leads into research of how to define and automatically extract topics from a meeting recording.

Furthermore, we will investigate into transcriptions from Annotation Graphs [2] into CMML.

Thirdly, it is not uncommon for a speaker in a meeting to refer to a previous agreement ("As we have agreed last week in topic 4,...") or a document ("Have a look at page 3 of this presentation..."). Our vision is to use speech processing and language technology to identify such references. This could be used to automatically create hyperlinks from meeting recordings to other digital media on the Web to truly create a Continuous Media Web.

7. REFERENCES

[1] annodex.net. Annodex: Open Standards for Annotating and Indexing Networked Media. http://www.annodex.net, 2003.

[2] Steven Bird and Mark Liberman. A Formal Framework for Linguistics Annotation. *speechcom*, 2000.

[3] Steve Cassidy. Generalising XPath for Directed Graphs. In *Proceedings of Extreme Markup Languages*, Montreal, Canada, August 2003.

[4] Steve Cassidy and Steven Bird. Querying Databases of Annotated Speech. In Maria E. Orlowska, editor, *Proceedings of the 11th Australasian Database Conference*, volume 22, pages 12–20, 2000.

[5] World Wide Web Consortium. XSL Transformations (XSLT) Version 1.0. http://www.w3.org/TR/xslt/.

[6] Dublin Core Metadata Initiative. Dublin Core Metadata Element Set, Version 1.1. http://dublincore.org/documents/2003/02/04/dces, February 2003.

[7] Nelson Morgan, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Adam Janin, Thilo Pfau, Elizabeth Shriberg, and Andreas Stolcke. The Meeting Project at ICSI. In *Proceedings of Human Language Technologies Conference*, San Diego, CAL, USA, March 2001.

[8] NIST. Rich Transcription 2004 Spring Meeting Recognition Evaluation. http://www.nist.gov/speech/tests/rt/rt2004/ spring/.

[9] Silvia Pfeiffer, Conrad Parker, and André Pang. Specifying Time Intervals in URI Queries and Fragments of Time–Based Web Resources (BCP) (work in progress). http://www.ietf.org/internet-drafts/draft-pfeiffer-temporal-fragments-02.txt, December 2003.

[10] Silvia Pfeiffer, Conrad Parker, and André Pang. The Annodex Annotation Format for Time–Continuous Bitstreams, Version 2.0 (work in progress). http://www.ietf.org/internet-drafts/draft-pfeiffer-annodex-01.txt, December 2003.

[11] Silvia Pfeiffer, Conrad Parker, and André Pang. The Continuous Media Markup Language (CMML), Version 2.0 (work in progress). http://www.ietf.org/internet-drafts/draft-pfeiffer-cmml-01.txt, December 2003.

[12] Silvia Pfeiffer, Conrad Parker, and André Pang. *Managing Multimedia Semantics*, Chapter: *Continuous Media Web: Hyperlinking, Search & Retrieval of Time–Continuous Data on the Web*. Idea Group, Inc., 2004. [accepted for publication].

[13] Silvia Pfeiffer, Conrad Parker, and Claudia Schremmer. Annodex: A Simple Architecture to Enable Hyperlinking, Search & Retrieval of Time–Continuous Data on the Web. In *Proc. 5th ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR)*, pages 87–93, Berkeley, CA, USA, November 2003. ACM.

[14] Silvia Pfeiffer and Claudia Schremmer. *MultiMedia Information Retrieval: metodologie ed esperienze internazionali di content–based retrieval per l'informazione e la documentazione*, Chapter: *Automated Annodexing of Meeting Recordings*. Associazione Italiana per la Documentazione Avanzata (AIDA), Roma 2004, Italy, March 2004.

[15] A Universal Transcription Format (UTF) Annotation Specification for Evaluation of Spoken Language Technology Corpora. http://www.nist.gov/speech/tests/bnr/bnews_99/utf-1.0-v2.ps, 1998. version 1.0.