

A Metadata Model for Capturing Presentations

Ralf Einhorn, Stephan Olbrich, Wolfgang Nejdil
Learning Lab Lower Saxony
University of Hannover
Expo Plaza 1, D-30539, Hannover, Germany
{einhorn,olbrich,nejdl}@learninglab.de

Abstract

This paper describes the design of a metadata model for capturing presentations developed as part of the VACE project (Video and Audio Capturing and Embedding). VACE is a modular, open, distributed framework for capturing presentations like lectures by using standard presentation and publishing tools for different media types. Different media formats can be used in one recording session in order to suit the needs of different presentation types, e. g. slides plus the talk of a lecturer.

Metadata are necessary to combine these media data in an efficient way. The combination of content based and synchronisation metadata is utilized for the integration of recorded material e. g. in web based learning systems, to provide navigation and search functions but can also be used for other post production purposes, e. g. video editing or DVD authoring.

1. Introduction, Motivation and Background

The goal of VACE is to provide a flexible framework for capturing presentations like lectures, courses or seminars primarily for the integration in repositories for tele-learning. While developing interactive multimedia



Figure 1. Actual presentation

learning material is still hard, recordings of actual presentations (which are existing anyway) can also be very useful to help students (Figure 1. Actual presentation). This cannot be accomplished, however, by simply putting a camera into a lecture hall.

1.1. Anatomy of a presentation

A usual presentation consists at least of a presenter (lecturer). In most cases the presenter will use tools for visualizing the content e. g. by using slides or a blackboard.

The difficult task is to generate a digital representation by conveying the original media without losing information. While it is clear that we have to create an audiovisual recording of the talk of the lecturer there are several ways to represent the content of the slides etc. To simplify the model in the following we talk about different continuous and discrete media streams.

When doing multiple recordings we get several different timelines that will be discussed later. We will also see that we can even improve the representation compared to the original presentation by adding certain navigational and random access functions the live audience does not have (see example in Figure 2. Datastreams of a presentation).

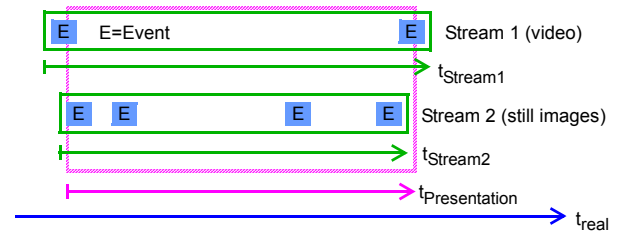


Figure 2. Datastreams of a presentation

1.2. Architecture of VACE

The following are the main characteristics of the VACE project.

Multimedia. To obtain an efficient representation of a real presentation the parallel recording of multiple media streams with different types and formats are utilized to meet the requirements of different presentation types. While the lecturer is captured with a continuous video- and audio recording, slides for example are stored temporally discrete as single images. While the A/V recording needs a higher temporal resolution a higher pixel resolution is needed for the slides. Of course the reuse of the original data would be the best solution. This is impossible for the

recording of the lecturer but can be obtained for some slide formats.

Distributed system. In order to capture different media in appropriate formats multiple systems have to be used for performance and compatibility reasons. Therefore real-time or off-line merge of the recorded data has to be enabled without losing synchronization.

Modular open system. VACE (other than monolithic systems) uses existing standard presentation, authoring and publication tools through appropriate interfaces. Additionally arbitrary presentation software can be integrated through generic image capturing. The VACE system provides the interface between the recording of real-life presentation and the production of a digital representation. Standard software systems are used for the representation.

Re-use of data. Data produced by the VACE system can be re-used for different purposes: different file formats (Real, Windows Media, Quicktime), video editing, DVD authoring. Of course the main target is the production of PC-based multimedia presentations e. g. based on SMIL.

Automatization. Most of the production flow should run automatically in order to minimize manual effort and to enable a quick publishing cycle. Thus the metadata should be generated during the actual presentation.

Combination of metadata. In VACE we distinguish between metadata in a conventional meaning (*content based metadata*) and metadata for storing the temporal sequence (*synchronization metadata*). A combination of both is used in VACE to enable indexing, navigation and searching within the parallel media streams. Both types of metadata are needed for the production of multimedia presentations to achieve a beneficial deployment e. g. within data bases and e-learning systems.

In a conventional way metadata is used for a whole resource (i. e. the presentation). Beside this, we need metadata for single passages. Hence we can make use of the fact that the – easy to generate – metadata of a lecture slide *also* describes the content of the synchronized A/V recording. As a result the user is not only able to find the slide but also the audiovisual explanation of the lecturer. Hence the heading "squirrel" of a slide as a textual metadata item can serve to find the associated aural explanation within the A/V data because the lecturer certainly talked about squirrels while showing this slide.

2. Evaluation of current metadata models

This chapter discusses the data models of related work like Dublin Core and SMIL relevant for VACE. After this section we will present and discuss the metadata elements for VACE.

2.1. Dublin Core Metadata Initiative (DCMI)

Dublin Core (DC) [1] offers a comprehensive classification for content based metadata. The model is not restricted to certain kinds of media but also does not provide special items for audiovisual media.

The Video Development Initiative (ViDe) published an application profile for digital video based on DC metadata [12]. Herein some DC classes have been extended. For example, the initiative suggests to add *video* and *animation* to the list of values for the DC element *type*. DC currently uses the common value *image* for all kinds of video.

The described elements are relevant for VACE and therefore could be adopted to our metadata scheme. In principle arbitrary content based metadata elements from DC can easily be added if considered useful.

Content. Description: certain subtypes like *table of contents* or *abstract*. ViDe complements a defined list of genres (e. g. containing *classroom lecture*). **Subject:** Keywords describing the topic and content of a resource. **Title:** In order to tag splitted video streams ViDe suggests to add *sequence* or *excerpt* to the title.

Intellectual Property. Creator: Synonymous to author. **Rights:** Copyright information in free form text.

Instantiation. Date: Date within the life cycle of a resource. DC uses the format described in [13] for date and time values. **Format:** Internet mediatype referenced in [3], corresponding to MIME type.

2.2. Synchronized Multimedia Integration Language (SMIL)

SMIL [14] defines a language for synchronized presentation of media streams and therefore can be regarded as one primary *target* format for presentations captured with the VACE system. Thus the usage of similar data structures during the recording was obviously useful. Because of the different application areas *recording* (VACE capturing) and *representation* (SMIL payout) a direct usage is not possible, however. While VACE focuses on data structures for the registration of real presentations, SMIL remains a format for the end product. An extensive SMIL implementation is part of the A/V streaming system from Real Networks [9].

Here are the SMIL elements (tags) and attributes relevant for the VACE metadata model.

The SMIL meta tag: The meta tag `<meta name="value" content="..." />` contains some basic content based metadata similar to the identical HTML tag. For *name* the tags *abstract*, *author*, *base*, *copyright* and *title* are defined. The *content* value may contain text or an URL.

The SMIL-ID: An ID can be assigned to each SMIL tag. This enables an unambiguous distinction of parallel media streams within a SMIL presentation and in a SMIL browser. There are two restrictions for values: Values are case-sensitive and the first character has to be a letter, colon or underscore.

Time formats in SMIL: SMIL offers two kinds of temporal statements (e. g. *start time* or *length*). Beside the *shorthand* variants like *1.5h*, *2.34min*, *5.6s* or *300.1ms* there is a standard format *hh:mm:ss.xy* with hours, minutes, seconds and hundredths of seconds.

Moreover there is a method for the synchronization of play-outs with the actual time called *wallclock-sync* using absolute time values from [13]. This mechanism may not be confused with the time values captured while recording.

2.3. MPEG-4, MPEG-7

While the MPEG-4 [5] standard describes mechanism for compositing and syncing media streams, the MPEG-7 [6] multimedia description schemes (DS) provide metadata structures for describing and annotating multimedia content. Most of the elements focuses on low-level attributes of A/V content like shape, color or motion an are intended to be extracted automatically. Some DSs are defined for content management: creation/production, media information and usage (rights).

The evolving MPEG standards focus on aspects of technical levels far beyond of the scope of this application. Adopting MPEG-7 would therefore result in a rather large overhead, and therefore was not considered for this project.

2.4. Other related projects

There are several projects with similar goals. Though most of them focus on different aspects.

- While out-of-the-box tools like Real Presenter [10], Microsoft Producer [4] and Lecturnity [2] are easy to use, they are tied to MS Powerpoint as presentation tool. The Real and Microsoft systems are restricted to vendor dependent A/V codecs. Because they run on a single machine they also lack of compatibility or performance problems for high-quality videos.

- Some tools use special applications for presentation, recording and publishing. While achieving a very exact representations even of the whiteboard content, authors and audience are forced to install specific software. [7]

- Other approaches use single media, i. e. wrap all data in a single high-quality video. Therefore efficiently navigation and search are not possible. [8]

3. A hierarchical metadata model

In this section we first discuss the various requirements for metadata of a distributed capturing system. Then we divide the presentation metadata into a structure of three hierarchical elements and define the single items.

3.1. Metadata requirements for VACE

For recording a live presentation all information necessary for the production of the digital reproduction (*representation*) have to be stored.

Beside the *media data* like the video recording or captured slides we need *metadata* primarily for the synchronization. Hence we need a data structure that reflects the original scenario. While the data is mainly needed for making multimedia presentations it seems to be reasonable to reuse existing multimedia presentations formats. Nevertheless the VACE data format is not meant to be used by the end user application.

3.1.1. Content based metadata. Content based metadata is needed especially for the indexing of the recorded media data. To implement interfaces for existing systems like on-line learning systems, standardized classifications like Dublin Core are preferred.

Global metadata. While adopting DC to VACE it has to be pointed out that DC is primarily designed for *single* resources like texts, images or (regarding the ViDE extension) videos. This kind of global metadata like *title*, *creator* or *rights* is used for the whole presentation.

Event related metadata. In addition to global metadata, elements are needed for the description of *fragments* of resources e. g. a single slide corresponding to an event. By combining this kind of content based metadata with synchronization metadata we are able to use search and navigation techniques known from textual media. Basically we need keywords describing the current content.

Linkage of metadata. While automatic indexing respectively metadata extraction from A/V media data files is difficult to implement it is easier and often more efficient to use metadata from an associated annotation data stream and thereby find the linked A/V content.

Furthermore the analysis of image data containing the slides used in the presentation can deliver more useful content based metadata e. g. by extracting keywords with character recognition software (OCR).

3.1.2. Synchronization metadata. For the recording the exact points of time (*timestamps*) have to be registered whenever the used media types change their states.

Media Types. There are basically two types of media relevant for capturing.

- For temporal *continuous media* like video recordings start and end time (resp. length) are sufficient. If more files or tapes are used for the recording of the same type of content we also need a unique identification of each part. Disregarding the potential jitter is a valid assumption for the used digital media formats.

- For temporal *discrete* media (e. g. a series of still images of screenshots) timestamp and content (resp. a reference to the content) have to be stored as well. We assume that the image (page, slide etc.) will be replaced by the next one so that an explicit length value (the duration of appearance) is not necessary. Otherwise a special *blank* item meets the requirement for this purpose.

Timelines. In a multimedia system we have to cope with several different bases for temporal information (compare the different timelines in Figure 2. Datastreams of a presentation).

As we have a *distributed* system working with several media streams we have to find a way to synchronize these streams. The basic idea of VACE is quite simple: using a common time base for the capturing process – the actual time.

- The real time serves as base for the recording. Thus we assume synchronized clocks on all systems involved. This can be achieved by using the Network Time Protocol [11].

- Each A/V recording again has its own timebase. A/V recordings on PCs (A/V files) usually start at zero. For video tapes also frame based timecode with randomly defined start values can be used.

- For the *representation* which is produced from the captured material the original time is of little relevance (as one item of the content based metadata) because the generated presentation uses a timeline defined by the author. For the users' point of view this *presentation timeline* starts at zero.

Time resolution. For the synchronization of the described application a resolution in the range of 1/100 to 1/10 seconds is considered sufficient.

For content based metadata usually the specification of date and time (in hours and minutes) is suitable. Audiovisual systems on the other hand need finer resolution but time values liberated from the real clock time. What we need is both: high resolution *and* real time.

Time format. The W3C describes a subset of time formats [13] defined in ISO 8601. The extended format with the highest resolution is CCYY-MM-DDThh:mm:ss.sTZD. T separates date from time. The time zone designator (TZD) can be Z (synonymous to UTC) or +hh:mm resp. -hh:mm. The number of digits for the decimal fraction of a second is not specified in [13]. This format was adopted by the XML Schema as *dateTime*

dateTime[15] with optional fractional seconds and with any number of digits after the decimal point. Appending the TZD is optional.

For the recording of a local presentation the timezone is of little meaning. To provide a simple alphanumeric sorting a fixed number of digits with a resolution of one millisecond can be used.

3.2. Definition of a hierarchical metadata model

The metadata that is needed can be separated in three hierarchical ordered elements (see Figure 3. Hierarchical presentation model and Figure 2. Datastreams of a presentation on page 1.

- Beside some global content based metadata from DC – which is valid for the whole recording – the **presentation** contains n_s ($n_s \geq 1$) media streams (called *streams*).

- A (media) **stream** is the representation of *one* media type, e. g. the A/V recording of the presenter and the series of slides is one stream each. When two video tapes are used for the recording of the presenter because of length limitations they count as one stream. On the other hand the additional recording of an experiment (e. g. in higher resolution) is another stream. A stream consists of its global data and n_e ($n_e \geq 1$) *events*.

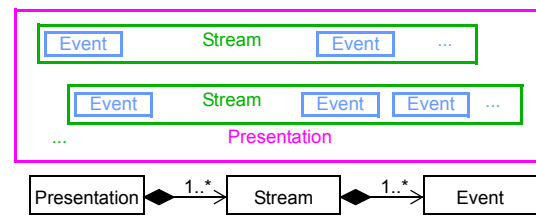


Figure 3. Hierarchical presentation model

- An **event** depicts the correlation between the media data and the time moment ("what happened when", e. g. the starting time of a video recording). There are several predefined event types depending on the *stream* type.

For each of these three elements *presentation*, *stream* and *event* data structures have to be defined. The following text describes the necessary sub-elements. Instead of using the SMIL tags the DC notation is used where available. Nevertheless these tags can easily be converted to similar SMIL statements (e. g. *copyright* instead of *rights*). The tags in italics are not included in DC or SMIL and therefore are introduced in VACE. XML is used as a data format and for each element the XML tag is given. Because the data mainly consist of lists, other data formats also can be used e.g. to fulfil the needs for streaming purposes. In the column "properties" the value type and format for each element is listed. For "dateTime" the W3C *dateTime* format is used with a fixed number of three digits for the decimal

fraction and leading zeros for each item (YYYY-MM-DDThh:mm:ss.xyz with xyz in ms, e. g. 2002-05-29T09:30:03.069).

Presentation. The *presentation* provides the container for the metadata of a recording. Beside the *stream* data some basic global informations are stored here. Other DC elements can easily be added.

The elements *starttime* and *endtime* determine the presentation timeline (see Figure 2. Datastreams of a presentation). All other elements contain free-form text without line breaks. Content based metadata within *presentation* is valid for the whole recording.

Table 1. Presentation metadata

XML-Tag	Description	Properties	mandatory
title	title of the presentation	text	yes
creator	author, creator	text	optional
rights	copyright information	text	optional
description	description of content	text	optional
subject	keywords, e. g. for search	text	optional
<i>starttime</i>	presentation start time	dateTime	yes
<i>endtime</i>	end time resp. length	dateTime	yes

Stream. A *stream* is the container for each media data stream. Different media types are identified by MIME-Types e. g. *video/mpeg1*. The content for the type *annotation* is exceptionally stored directly within the meta data and therefore does not need any additional media file. The list of MIME-types in [3] unfortunately does not contain all necessary types, e. g. *application/vnd.rn-realmedia* for popular RealMedia files is missing. Some extensions can also be found in the ViDE list [12].

Table 2. Stream metadata

XML-Tag	Description	Properties	mandatory
format	media type (audio, video, A/V, URL, image, annotation)	MIME type/subtype	yes
id	unambiguous identification	text	yes
<i>timeoffset</i>	$t_{offset} = t_{real} - t_{stream}$	timeDuration	yes
description	description of content	text	optional
subject	keywords, e. g. for search	text	optional
<i>filebase</i>	common path for media data	text	optional
<i>filepattern</i>	filename mask ("capture*.png")	text	optional

Each *stream* has a *id* that is unique within each presentation. The *filebase*-tag facilitates the consistent data storage and holds the information where the files (which names are given by *filepattern*) are stored e. g. for post-production. Both elements are used for implementation reasons to achieve easy file handling. In principle these values could be added to each filename. They are not used for the stream type *annotation*. The actual filenames of media data files are not contained within *stream* but in each *event* belonging to the stream.

Event. The central element of an *event* is its *time*. The size of a time window for a presentation normally ranges

from a couple of minutes to some hours. The necessary resolution is about some milliseconds. To simplify alphanumeric sorting, date and time are combined in one element. In most cases the date information is not needed because presentations do not spread over days. For time windows around midnight this enables a continuous timeline.

Table 3. Event metadata

XML-Tag	Description	Properties	mandatory
type	type of event	predefined type	yes
time	time of event in real time	dateTime	yes
title	title	text	optional
description	description of content	text	optional
subject	keywords, e. g. for search	text	optional
<i>file</i>	URL or name of media file	text	optional
<i>timebase</i>	starting time (default: zero)	timeDuration	optional

In continuous A/V recordings *timeoffset* matches the (first) starting time. Thus, it is possible to display the file-based timeline within the VACE editor.

The name of the media file is hold in *file*, e. g. the image file of a screenshot. Location and format of *file* are stored in the global *stream* data. Alternatively complete (fully qualified) URLs can be used.

Eventtypes. For each different *stream* type several different *types* for events are defined. For *video/* start* and *end* (where *end* implicitly holds the length) for *image/* img*, for *text/* href*, and for *annotations* HTML tags to denote different levels of hierarchy: *h1, h2, h3, h4, p*.

4. Implementation

The implementation of the VACE system comprises the development of a core component plus certain interfaces and filters for existing software (i. e. presentation tools and A/V encoder). The VACE Editor currently has prototype state. It allows to create, read and merge VACE metadata files in XML format. All elements can be edited. Annotations and events can also be entered in real time i. e. during a presentation (Figure 4. VACE Editor).

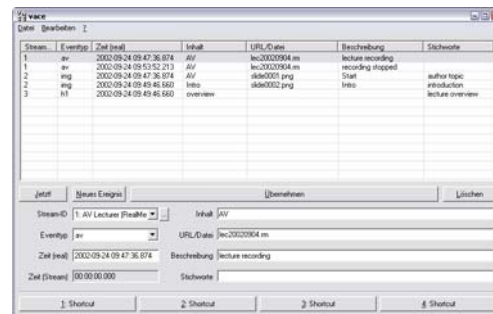


Figure 4. VACE Editor

5. Conclusions and Future Work

The definition of the metadata model (re-using existing standards as much as possible) is a first step towards a framework for the efficient distributed capturing of presentations. The VACE framework facilitates the post-production of live recordings. Capturing tools can be used platform- and vendor-independent. Future work will consist of implementing VACE compliant generic capturing tools and setting up interfaces to common presentation, post-production and publishing tools.

6. References

- [1] Dublin Core Metadata: <http://dublincore.org/>
- [2] Lecturnity: <http://www.im-c.de>
- [3] Media Types: <http://www.iana.org/assignments/media-types/>, <http://www.isi.edu/in-notes/iana/assignments/media-types/media-types>
- [4] Microsoft Producer for Powerpoint: <http://microsoft.com/office/powerpoint/producer/>
- [5] MPEG-4 Overview: ISO/IEC JTC1/SC29/WG11 N4668, March 2002, Editor: Koenen R
- [6] MPEG-7 Overview: ISO/IEC JTC1/SC29/WG11 N4980, Klagenfurt, July 2002, Editor: Martínez J M (UPM-GTI, ES)
- [7] Müller R, Ottmann T: The "Authoring on the Fly" system for automated recording and replay of (tele)presentations, *Multimedia Systems*, *Multimedia Systems* 8 Issue 3 (2000) pp 158-176
- [8] Naegele-Jackson S, Gräve M, Eschbaum N, Holleczeck P, "Distributed TV Productions and Video-on-Demand Services at Universities" (UniTV), TERENA Networking Conference 2000, Lisbon, Portugal, 05/2000, <http://www.uni-tv.net>
- [9] Real Networks: RealSystem iQ Production Guide, <http://service.real.com/help/library/guides/realone/ProductionGuide/PDF/ProductionGuide.pdf>
- [10] Real Presenter: <http://www.realnworks.com/products/presenterone/>
- [11] RFC 1305: Network Time Protocol (Version 3), Specification, Implementation and Analysis, David L. Mills, 3/1992
- [12] ViDe User's Guide: Dublin Core Application Profile for Digital Video, http://www.vide.net/workgroups/videoaccess/resources/vide_dc_userguide_20010909.pdf, Editors: Agnew G, Kniesner D
- [13] W3C: Date and Time Formats, Note, <http://www.w3.org/TR/1998/NOTE-datetime-19980827>, Wolf M, Wicksteed C
- [14] W3C: Synchronized Multimedia Integration Language (SMIL 2.0), W3C Recommendation 07 August 2001, <http://www.w3.org/TR/2001/REC-smil20-20010807/>
- [15] W3C: XML Schema Part2: Datatypes, <http://www.w3.org/TR/xmlschema-2/#dateTime>, W3C Recommendation 02 May 2001, Editors: Biron P V, Malhotra A