Deliverable 1.1.2
Report on MediaMixer core technology set
Update 1, 08 May 2013

Coordinator: Jan Thomsen, Condat AG
With contributions from: Media Mixer consortium
Quality Reviewer: Benoit Huet, Eurecom

Editor: Jan Thomsen, Condat AG
Other Authors: Raphael Troncy (EURECOM), Vasileios Mezaris (CERTH), Martin J. Dow (ACUITY), Lyndon Nixon (STI), Roberto Garcia (UDL), Rolf Fricke (CONDAT)
Deliverable nature: Report (R)
Dissemination level: Public (PU)
Contractual delivery date: 30 April 2013
Actual delivery date: 30 April 2013
Version: 1.0
Total number of pages: 59
Keywords: Media Fragments, Media Mixing Tools, Media Analysis, Media Enrichment, Media Retrieval, Digital Rights Management
Executive summary

In order to foster the usage of annotated media fragments within the creative industries, organizations and communities MediaMixer is setting up a community for enabling discussions, creating use cases and make known the technologies which are available or being developed right now. This report describes the media fragment related core technologies, standards and toolsets which cover the whole life cycle of media fragments.

Media Fragment Specification, Servers and Clients describes the details of the W3C Media Fragment 1.0 standard and media delivery servers as well as media players and frameworks which support this standard.

Media Fragment Creation describes the video analysis techniques which are applied to media resources in order to create media fragments by performing different analysis procedures like automatic speech recognition, face recognition or concept redetection.

Media Fragment Annotation describes the different technologies and standard to annotate media fragments, i.e. assign recognized named entities to media fragments, which are also the basis for later enrichment to further link these media fragments to Linked Open Data sources or related multimedia or other content.

Media Fragment Management describes what procedures, tools and technologies can be used in order to implement an effective management of media fragments.

Media Fragments Rights Management and Negotiation motivates the very sensitive topic of digital rights management with respect to media fragments, a topic which is absolutely essential with respect to a industry-wide adoption of media fragments.

Media Fragments Lifecycle, finally, gives an overview of the whole process of media fragment usage from creation unto publishing and consumption.

A conclusion section points to some further issues which need to be tackled by the media fragment community in the future.
Document Information

<table>
<thead>
<tr>
<th>IST Project Number</th>
<th>Acronym</th>
<th>MEDIAMIXER</th>
</tr>
</thead>
<tbody>
<tr>
<td>FP7 - 318101</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Full Title  Community set-up and networking for the reMIXing of online MEDIA fragments

Project URL  http://www.mediamixer.eu

Document URL

EU Project Officer  Sophie REIG

Deliverable Number  1.1.2
Title  Report on MediaMixer core technology set

Work Package Number  1
Title  MediaMixer Online Portal

Date of Delivery
Contractual M6
Actual M6

Status  version 1.0
final X

Nature  prototype ☐ report X dissemination ☐

Dissemination level  public X consortium ☐

Authors (Partner)
Jan Thomsen (CONDAT), Raphael Troncy (EURECOM), Vasileios Mezaris (CERTH), Lyndon Nixon (STI), Martin J. Dow (ACUITY), Roberto Garcia (UDL), Rolf Fricke (CONDAT)

Responsible Author
Name  Jan Thomsen
E-mail  jan.thomsen@condat.de
Partner  Condat AG
Phone  +49 30 3949 1181

Abstract
This Deliverable summarizes the current state of the art and technology concerning media fragments, covering the specification of the W3C Media Fragment URI 1.0 standard, the video analysis processes which lead to the creation of media fragments, annotation and enrichment, DRM, management and lifecycle of media fragments.

Keywords  Media Fragments, Media Mixing Tools, Media Analysis, Media Enrichment, Media Retrieval, Digital Rights Management

Version Log

<table>
<thead>
<tr>
<th>Issue Date</th>
<th>Rev. No.</th>
<th>Author</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>26.03.2013</td>
<td>0.1</td>
<td>Rolf Fricke</td>
<td>Structure and initial content</td>
</tr>
<tr>
<td>05.04.2013</td>
<td>0.2</td>
<td>Roberto García González</td>
<td>Added section on rights management</td>
</tr>
<tr>
<td>10.04.2013</td>
<td>0.3</td>
<td>Vasileios Mezaris</td>
<td>Added section on media fragment creation</td>
</tr>
<tr>
<td>12.04.2013</td>
<td>0.4</td>
<td>Rolf Fricke</td>
<td>Inclusion of workflow and SME section</td>
</tr>
<tr>
<td>25.04.2013</td>
<td>0.5</td>
<td>Jan Thomsen</td>
<td>Overall edition, inclusion of media fragments lifecycle</td>
</tr>
<tr>
<td>28.04.2013</td>
<td>0.6</td>
<td>Raphael Troncy</td>
<td>Added section on media fragment specification, servers and clients</td>
</tr>
<tr>
<td>29.04.2013</td>
<td>0.7</td>
<td>Jan Thomsen</td>
<td>Final editing and adding of missing parts</td>
</tr>
<tr>
<td>29.04.2013</td>
<td>0.8</td>
<td>Martin Dow</td>
<td>added section on Media Fragment Management</td>
</tr>
<tr>
<td>30.04.2013</td>
<td>1.0</td>
<td>Jan Thomsen</td>
<td>Integrated comments from QA; ready for submission</td>
</tr>
</tbody>
</table>
# Table of Contents

EXECUTIVE SUMMARY .................................................................................................................. 2

DOCUMENT INFORMATION ............................................................................................................. 3

TABLE OF CONTENTS ................................................................................................................... 4

LIST OF FIGURES .......................................................................................................................... 6

1 INTRODUCTION ......................................................................................................................... 7

2 MEDIA FRAGMENT SPECIFICATION, SERVERS AND CLIENTS .................................................. 9

   2.1 MEDIA FRAGMENT URIs SPECIFICATION ......................................................................... 9
     2.1.1 Media Resource Model ................................................................................................. 9
     2.1.2 Fragment Dimensions .................................................................................................. 9
     2.1.3 URI Fragments vs. URI Queries .................................................................................... 10

   2.2 MEDIA FRAGMENT IMPLEMENTATIONS .......................................................................... 11
     2.2.1 Ninsuna Media Delivery Platform ............................................................................... 11
     2.2.2 Web browsers support............................................................................................... 12
     2.2.3 mediafragment.js Library ........................................................................................... 14
     2.2.4 xywh.js Library ........................................................................................................... 15
     2.2.5 Synote Media Fragment Player (SMFP) ....................................................................... 16
     2.2.6 Ligne de Temps ........................................................................................................... 16

3 MEDIA FRAGMENT CREATION ................................................................................................... 18

   3.1 SHOT SEGMENTATION ..................................................................................................... 18
     3.1.1 Purpose ........................................................................................................................ 18
     3.1.2 Material types .............................................................................................................. 18
     3.1.3 Methods ....................................................................................................................... 18
     3.1.4 Results ........................................................................................................................ 19
     3.1.5 Constraints .................................................................................................................. 20
     3.1.6 Implementation and Licensing ................................................................................... 20
     3.1.7 Remaining challenges ............................................................................................... 20

   3.2 VIDEO CONCEPT DETECTION ......................................................................................... 20
     3.2.1 Purpose ........................................................................................................................ 20
     3.2.2 Material types .............................................................................................................. 20
     3.2.3 Methods ....................................................................................................................... 21
     3.2.4 Results ........................................................................................................................ 22
     3.2.5 Constraints .................................................................................................................. 22
     3.2.6 Implementation details and licensing .......................................................................... 22
     3.2.7 Remaining challenges ............................................................................................... 23

   3.3 OBJECT RE-DETECTION ................................................................................................. 23
     3.3.1 Purpose ........................................................................................................................ 23
     3.3.2 Material types .............................................................................................................. 23
     3.3.3 Methods ....................................................................................................................... 24
     3.3.4 Results ........................................................................................................................ 24
     3.3.5 Constraints .................................................................................................................. 25
     3.3.6 Implementation details and licensing .......................................................................... 25
     3.3.7 Remaining challenges ............................................................................................... 25

4 MEDIA FRAGMENT ANNOTATION ............................................................................................... 26

   4.1 EXISTING MEDIA DESCRIPTION MODELS .................................................................... 26
   4.2 W3C MEDIA ONTOLOGY AND API ................................................................................. 28
5 MEDIA FRAGMENT MANAGEMENT .......................................................... 35
5.1 PURPOSE .......................................................................................... 35
5.2 BACKGROUND DRIVERS AND MOTIVATING SCENARIOS ............... 35
  5.2.1 Emergent economics of preservation and access .......................... 35
  5.2.2 Heterogeneity in tooling and delivery networks: towards cloud-based solutions .......................................................... 36
  5.2.3 From files to digital objects .......................................................... 37
  5.2.4 Collection management and archive management ....................... 37
5.3 MATERIAL TYPES AND METHODS ............................................... 38
  5.3.1 Digital asset management system based on Fedora Commons ........ 38
  5.3.2 Identifier Framework .................................................................. 38
  5.3.3 Digital Object Model ................................................................. 39
  5.3.4 Content Model Architecture (CMA) and Behaviours ...................... 40
  5.3.5 Semantic Web Capabilities .......................................................... 40
  5.3.6 Content Protection Measures ...................................................... 41
  5.3.7 Support for rights ..................................................................... 41
  5.3.8 Provenance and Trust .................................................................. 41
  5.3.9 High Level Storage API ............................................................... 42
5.4 CONSTRAINTS .................................................................................. 42
  5.4.1 Facilities outside core MediaMixer concerns .................................. 42
  5.4.2 Summary list of Fedora Commons features ................................... 42
5.5 REMAINING CHALLENGES ............................................................ 42
6 MEDIA FRAGMENTS RIGHTS DESCRIPTIONS AND NEGOTIATIONS ............ 44
  6.1 PURPOSE ...................................................................................... 44
  6.2 SCENARIO MOTIVATING COPYRIGHT MANAGEMENT OF MEDIA FRAGMENTS .................................................. 44
  6.3 MATERIAL TYPES ....................................................................... 45
  6.4 METHODS ..................................................................................... 45
  6.5 RESULTS ........................................................................................ 46
  6.6 ALTERNATIVE TOOLS ................................................................. 47
  6.7 OS/COSTS ...................................................................................... 48
  6.8 REMAINING CHALLENGES .......................................................... 48
7 MEDIA FRAGMENT LIFECYCLE ............................................................... 50
  7.1 ANALYSIS ..................................................................................... 50
    7.1.1 Media Resource Ingestion .......................................................... 50
    7.1.2 Media Resource Analysis ............................................................ 51
    7.1.3 Media Fragment Generation ...................................................... 51
  7.2 METADATA AGGREGATION ......................................................... 52
    7.2.1 Metadata Conversion ................................................................. 52
    7.2.2 Annotation and Enrichment ........................................................ 53
    7.2.3 Editing ..................................................................................... 53
  7.3 PUBLISHING AND CONSUMPTION ............................................. 54
8 CONCLUSIONS .................................................................................... 56
9 REFERENCES ....................................................................................... 57
List of figures

Figure 1: Ninsuna Media Fragments Player: example of spatio-temporal media fragment ..................................12
Figure 2: Implementation of Media Fragments URI in Mozilla Firefox 9+ ..............................................................13
Figure 3: Implementation of Media Fragments URI in Chrome (Webkit) .................................................................14
Figure 4: Spatial dimension example of Media Fragments URI using xywh.js library .............................................15
Figure 5: Synote Media Player screenshot .............................................................................................................16
Figure 6: Implementation of Media Fragments URI in LigneDeTemp .................................................................17
Figure 7: Workflow of the shot segmentation algorithm, described in [LinkedTV] .................................................19
Figure 8: Video shot segmentation example ..........................................................................................................20
Figure 9: Workflow of the concept detection technique, described in [MGS+12] .......................................................21
Figure 10: Performance of three different concept detection systems .................................................................22
Figure 11: Example of video concept detection output ............................................................................................23
Figure 12: Workflow of the object re-detection algorithm .......................................................................................24
Figure 13: Object of interest (up left) and detected appearances of it .................................................................25
Figure 14: Comparison of 4 MPEG7 ontological descriptions of the same image region ....................................27
Figure 15: MediaOntology annotation ...................................................................................................................28
Figure 16: High level view of ConnectME ontology structure ...............................................................................31
Figure 17: LinkedTV ontology components ........................................................................................................32
Figure 18: Screenshot of annotator .........................................................................................................................33
Figure 19: OAIS Functional Model .......................................................................................................................36
Figure 20: Fedora Digital Object ............................................................................................................................39
Figure 21: The creation model provided by the Copyright Ontology .......................................................................45
Figure 22: Illustration of reasoner classification service for checking if copyright dispute is supported by existing license deals ..................................................................................................................47
Figure 23: Overview of the Media Fragment Lifecycle ..........................................................................................50
Figure 24: Overview of the Media Resource Analysis process ............................................................................50
Figure 25: Overview of the metadata aggregation process ...................................................................................52
Figure 26: Screenshot Condat SmartMediaEngine ............................................................................................54
1 Introduction

One main objective of the MediaMixer project is to evaluate the current workflow of fragment handling, to identify shortcomings and derive measures to enable a further adoption of media fragment exchange. Currently, in the media sector several tools already allow to create and exchange smaller video clips, but only very few of them already support the MF URI 1.0 specification. Despite this situation, it is already possible, to exchange and reuse video clips from different tools, but this usually requires to consider different video coding and annotation standards. Therefore, the access and exchange of fragments needs the management of different coding/decoding components or transformation tools. One objective of the MediaMixer project will be to facilitate the handling of media fragments through a more seamless workflow by a common use of the MF URI standard. This requires that the MF URI 1.0 standard is supported by the interfaces and parameters of the involved tools and applications for media fragment access and exchange.

Workflow
The evaluation of the conditions for an adoption of the media fragment standard requires to consider tools, frameworks and specifications needed along the workflow, which support the actors to provide, retrieve and reuse media fragments. The workflow starts with the creation of media fragments from raw materials, footage or any other existing videos. The first task is the extraction of useful parts from the footage. Each of these parts needs to be annotated, in order to allow them to be manageable and to be able to be found in the following process steps. Subsequently the fragments have to be stored and managed. For the provisioned users, functions for navigation and retrieval have to be provided. They also need the information about rights and costs, in order to decide which of the fragments best fits the current need.

User groups
The following user groups are involved in the workflow to provide or demand media fragments:

- Editors: users creating or updating content, e.g. for news production or creation of videos;
- Managing editors: responsible for the resulting product
- Rights department: management of rights, which may also include the negotiation of rights and costs for the specific use of materials with a certain target group and time frame
- Archive: management of long term storage of materials, which may include classification and annotation tasks
- Purchasing: payment for sold or bought materials

Domains
The tools and documentation of the following domains are examined:

- Standards, Specifications or best practices for media fragments
- Tool sets, frameworks, libraries, ontologies
  - from the MediaMixer community
  - free and Open Source tools

Different level of maturity for media fragment support
The MF URI 1.0 specification is a very new technology, which has been published as a W3C-recommendation in 2012. Therefore only a few tools already support this standard. Each domain of the workflow has a different level of maturity regarding the integration of the media fragment standards. For some domains, such as the MF generation and annotation, there are already some examples available. For others, like asset or rights management, which are quite complex applications implementing established workflows and are tightly connected to large media asset management systems or content management systems, there is nearly no support of the W3C specification available.

The outline of the Deliverable is as follows:
Chapter 2 Media Fragment Specification, Servers and Clients describes the details of the W3C Media Fragment 1.0 standard and media delivery servers as well as media players and frameworks which support this standard.

Chapter 3 Media Fragment Creation describes the video analysis techniques which are applied to media resources in order to create media fragments by performing different analysis procedures like automatic speech recognition, face recognition or concept redetection.

Chapter 4 Media Fragment Annotation describes the different technologies and standard to annotate media fragments, i.e. assign recognized named entities to media fragments, which are also the basis for later enrichment to further link these media fragments to Linked Open Data sources or related multimedia or other content.

Chapter 5 Media Fragment Management describes what procedures, tools and technologies can be used in order to implement an effective management of media fragments.

Chapter 6 Media Fragments Rights Management and Negotiation motivates the very sensitive topic of digital rights management with respect to media fragments, a topic which is absolutely essential with respect to a industry-wide adoption of media fragments.

Chapter 7 Media Fragments Lifecycle, finally, gives an overview of the whole process of media fragment usage from creation unto publishing and consumption.

A conclusion section points to some further issues which need to be tackled by the media fragment community in the future.
2 Media Fragment Specification, Servers and Clients

Video clips on the World Wide Web (WWW) used to be treated as "foreign" objects as they could only be embedded using a plugin that is capable of decoding and interacting with these clips. The HTML5 specification is a game changer and all of the major browser vendors now support the newly introduced `<video>` and `<audio>` elements. However, in order to make video clips accessible in a transparent way, it needs to be as easily linkable as a simple HTML page. In order to share or bookmark only the interesting parts of a video, we should be able to link into or link out of this time-linear media resource. If we want to further meet the prevailing accessibility needs of a video, we should be able to dynamically choose our preferred tracks that are encapsulated within this video resource, and we should be able to easily show only specific regions-of-interest within this video resource. And last but not least, if we want to browse or scan several video resources based on (encapsulated) semantics, we should be able to master the full complexity of rich media by also enabling standardised media annotation. Note that we can generalize the above observations to other media, such as audio resources. This way, media resources truly become first-class citizens on the Web.

The mission of the W3C Media Fragments Working Group (MFWG), which is part of W3C’s Video in the Web activity¹, is to provide a mechanism to address media fragments on the Web using Uniform Resource Identifiers (URIs). The objective of the proposed specification is to improve the support for the addressing and retrieval of sub-parts of so-called media resources (e.g. audio, video and image), as well as the automated processing of such sub-parts for reuse within the current and future Web infrastructure. Example use cases are the bookmarking or sharing of excerpts of video clips with friends in social networks, the automated creation of fragment URIs in search engine interfaces by having selective previews, or the annotation of media fragments when tagging audio and video spatially and/or temporally. Throughout this chapter, we will give various examples of tools and applications that use Media Fragments URI specification, in order to illustrate all those possibilities.

2.1 Media Fragment URIs Specification

2.1.1 Media Resource Model

We assume that media fragments are defined for "time-linear" media resources, which are characterised by a single timeline. Such media resources usually include multiple tracks of data all parallel along this uniform timeline. These tracks can contain video, audio, text, images, or any other time-aligned data. Each individual media resource also contains control information in data headers, which may be located at certain positions within the resource, either at the beginning or at the end, or spread throughout the data tracks as headers for those data packets. There is also typically a general header for the complete media resource. To comply with progressive decoding, these different data tracks may be encoded in an interleaved fashion. Normally, all of this is contained within one single container file.

2.1.2 Fragment Dimensions

Temporal Axis. The most obvious temporal dimension denotes a specific time range in the original media, such as "starting at second 10, continuing until second 20". Temporal clipping is represented by the identifier t, and specified as an interval with a begin and an end time (or an in-point and an out-point, in video editing terms). If either or both are omitted, the begin time defaults to 0 second and the end time defaults to the end of the entire media resource. The interval is considered half-open: the begin time is part of the interval whereas the end time on the other hand is the first time point that is not part of the interval. The time units that can be used are Normal Play Time (npt), real-world clock time (clock), and SMPTE timecodes. The time format is specified by name, followed by a colon, with npt: being the default. Some examples are:

¹ [http://www.w3.org/2008/WebVideo/Activity.html](http://www.w3.org/2008/WebVideo/Activity.html)
t=npt:10,20 → results in the time interval [10,20[

Spatial Axis. The *spatial dimension* denotes a specific spatial rectangle of pixels from the original media resource. The rectangle can either be specified as pixel coordinates or percentages. A rectangular selection is represented by the identifier xywh, and the values are specified by an optional format pixel or percent: (defaulting to pixel) and 4 comma-separated integers. These integers denote the top left corner coordinate (x,y) of the rectangle, its width and its height. If percent is used, x and width should be interpreted as a percentage of the width of the original media, and y and height should be interpreted as a percentage of the original height. Some examples are:

\[
\begin{align*}
xywh=160,120,320,240 & \rightarrow \text{results in a 320x240 box at x=160 and y=120} \\
xywh=pixel:160,120,320,240 & \rightarrow \text{results in a 320x240 box at x=160 and y=120} \\
xywh=percent:25,25,50,50 & \rightarrow \text{results in a 50%x50% box at x=25% and y=25%}
\end{align*}
\]

Track Axis. The *track dimension* denotes one or multiple tracks, such as "the English audio track" from a media container that supports multiple tracks (audio, video, subtitles, etc). Track selection is represented by the identifier track, which has a string as a value. Multiple tracks are identified by multiple name/value pairs. Note that the interpretation of such track names depends on the container format of the original media resource as some formats only allow numbers, whereas others allow full names. Some examples are:

\[
\begin{align*}
\text{track}=1&\&\text{track}=2 & \rightarrow \text{results in only extracting track '1' and '2'} \\
\text{track}=\text{video} & \rightarrow \text{results in only extracting track 'video'} \\
\text{track}=\text{Kids}\%\text{Video} & \rightarrow \text{results in only extracting track 'Kids Video'}
\end{align*}
\]

Named Axis. The *named dimension* denotes a named section of the original media, such as "chapter 2". It is in fact a semantic replacement for addressing any range along the aforementioned temporal axis. Name-based selection is represented by the identifier id, with again the value being a string. Percent-encoding can be used in the string to include unsafe characters (such as a single quote). Interpretation of such strings depends on the container format of the original media resource. As with track selection, determining which names are valid requires knowledge of the original media resource and its media container format.

\[
\begin{align*}
\text{id}=1 & \rightarrow \text{results in only extracting the section called '1'} \\
\text{id}=\text{chapter}-1 & \rightarrow \text{results in only extracting the section called 'chapter-1'} \\
\text{id}=\text{My}\%\text{Kids} & \rightarrow \text{results in only extracting the section called 'My Kids'}
\end{align*}
\]

Combined Dimensions. As the temporal, spatial, and track dimensions are logically independent, they can be combined where the outcome is also independent of the order of the dimensions. As such, the following fragments should be byte-identical:

- http://example.com/video.ogv#t=10,20&track=vid&xywh=pixel:0,0,320,240
- http://example.com/video.ogv#track=vid&xywh=0,0,320,240&t=npt:10,20
- http://example.com/video.ogv#xywh=0,0,320,240&t=smpte:0:00:10,0:00:20&track=vid

2.1.3 URI Fragments vs. URI Queries

Without entering too much into details, the main difference between a URI query and a URI fragment is that a URI query creates a completely new resource having no relationship whatsoever with the resource it is created from, while a URI fragment delivers a secondary resource that relates to the primary resource. As a consequence, URI query created resources cannot be mapped byte-identical to their parent resource (this notion does not even exist), and are thus considered a re-encoded segment. The use of URI fragments is preferable over the use of URI queries since byte range requests are an inherent part of HTTP including the caching proxy infrastructure, while providing a query mechanism requires extending the server software. In the case of playlists composed of media fragment
resources, the use of URI queries (receiving a completely new resource instead of just byte segments from existing resources) could be desirable since it does not have to deal with the inconvenience of the original primary resources - its larger file headers, its longer duration, and its automatic access to the original primary resources.

2.2 Media Fragment Implementations

We described in the previous section the Media Fragment URI syntax. We present now how this specification can be used for implementing applications that deal with spatial and temporal media fragments in the Web, and how those applications bring new features to the world of multimedia and offer the users a new hypervideo experience.

2.2.1 Ninsuna Media Delivery Platform

Nisuna Media Delivery\(^2\) is a Model-driven media platform for multimedia content adaptation and delivery. Its basic design is inspired by the principles of XML-driven content adaptation techniques, while its final design and the implementation thereof are based on Semantic Web technologies such as the Resource Description Framework (RDF), Web Ontology Language (OWL), and SPARQL Protocol And RDF Query Language (SPARQL).

A tight coupling exists between the design of this media delivery platform and the proposed model for describing structural, content, and scalability information of multimedia bitstreams. This model, implemented by making use of OWL, provides support for a seamless integration of the adaptation operations and semantic metadata. Therefore, it enables the definition of adaptation operations on a higher level (i.e., based on the model). Furthermore, when existing coding formats are mapped to this model, they can be adapted in a format-independent way. Because the platform is fully based on Semantic Web technologies, advanced queries can be constructed by a client application and efficiently processed by the platform. This way, fully customized search applications can be developed to interact with the platform while searching for specific parts of the media content. Because media resources can be annotated down to the scene level, it is possible to only return the relevant media fragments that correspond to the search query (instead of the full media resource). Here is where the Media Fragment URI specification plays its role.

For testing this possibilities, a W3C Media Fragments Player 1.0 has been developed. There are two different versions: on the one hand, the Flash-based media fragments player\(^3\) works with codecs supported by Flash. Further, both HTTP and RTMP can be used as access protocol. On the other hand, the HTML5-based media fragments player\(^4\) is able to decode media formats that are supported by the browser the user is using. An example of those two players is shown in Figure 1.

\(^2\) [http://ninsuna.elis.ugent.be/ModelDrivenMediaDelivery](http://ninsuna.elis.ugent.be/ModelDrivenMediaDelivery)

\(^3\) [http://ninsuna.elis.ugent.be/MFPlayer/flash](http://ninsuna.elis.ugent.be/MFPlayer/flash)

Figure 1: Ninsuna Media Fragments Player: example of spatio-temporal media fragment.

On the server side, both the time and track fragment axes are supported by the media delivery platform. Media segments can be requested by using the query parameter and/or through the HTTP range header. Also, no transcoding is applied to create the media segments; more specifically, all segments are extracted from the original media resource. In addition, Ninsuna provides a W3C Media Fragments Validation Service, which allows external tools and users to syntactically validate Media Fragment URIs 1.0. This media fragments parser and validator is also available as a standalone Java program: MFV.jar. Its usage is as simple as run the command: java -jar MFV.jar <mediafragment>.

2.2.2 Web browsers support

Mozilla Firefox Support. Firefox has been an early implementor of the Media Fragments URI specification. It has been officially part of the browser since the version 9 released on December 20, 2011. At the moment, it supports only the temporal dimension and does not save bandwidth.

Figure 2 shows a demonstration of the native HTML5 player of Firefox playing only a fragment of an audio file encoded in WebM. The source code of this page reads:

[Source code]

5 http://ninsuna.elis.ugent.be/MFValidationService/resources/MFV.jar
6 See the announcement at http://lists.w3.org/Archives/Public/public-media-fragment/2011Nov/0017.html
<video id="v" src="AudioAPI.webm#t=50,100"
onloadedmetadata="update()" onpause="update()"
onplay="update()" onseeked="update()" controls>
</video>

**Temporal Media Fragment Test Builds**

Media fragment support has landed in the Firefox Nightly build.

**Example**

![Calculate Audio Spectrum (FFT) in JS](image)

*Figure 2: Implementation of Media Fragments URI in Mozilla Firefox 9+*

**WebKit Support.** WebKit implements the Media Fragments URI specification since the version: 528+ (Nightly build)\(^7\). As in Firefox, it supports only the temporal dimension and does not save bandwidth. The lead developer of this feature has been Eric Carlson, from Apple.

Figure 3 shows a demonstration of the native HTML5 player of Webkit in Chrome playing only a fragment of an audio file encoded in mp4. The source code is the following:

```html
... <video id="video_query1a" controls style="width: 238px; height: 230px">
</video>
```

\(^7\) See the bug closed at [https://bugs.webkit.org/show_bug.cgi?id=65838](https://bugs.webkit.org/show_bug.cgi?id=65838)
Figure 3: Implementation of Media Fragments URI in Chrome (Webkit)

2.2.3 mediafragments.js Library

The library mediafragments.js allows to easily parse MediaFragments URI, decomposing the entire chain into the different logic parts that are considered in the specification. The source code can be downloaded from GitHub\(^8\). It has been developed by Thomas Steiner and released under the CC0 1.0 Universal (CC0 1.0) license.

As an example, this is the result obtained from this library when parsing [https://github.com/tomayac/Media-Fragments-URI](https://github.com/tomayac/Media-Fragments-URI) as a Media Fragments URI:

[Query]:
* t:
  [ - value: clock:2011-10-01T23:00:45.123Z,2011-10-01T23:00:45.123Z
    - unit: clock
    - start: 2011-10-01T23:00:45.123Z
    - end: 2011-10-01T23:00:45.123Z
  ]

[Hash]:
* xywh:
  [ - value: pixel:10,10,30,30

\(^8\) [https://github.com/tomayac/Media-Fragments-URI](https://github.com/tomayac/Media-Fragments-URI)
2.2.4 xywh.js Library

The library xywh.js implements the spatial media fragments dimension of the W3C Media Fragments URI specification as a polyfill. For more information about what is and how to use a polyfill, see http://remysharp.com/2010/10/08/what-is-a-polyfill/. The code is available on GitHub9.

The usage is as simple as including xywh_min.js file in a Web application. The polyfill will run when the load event fires. Additionally, a function mediaFragments.apply() is exposed, which can be used to apply media fragments of dynamically added media items. For the mark-up, one should create media items such as:

```
<img src="kitten.jpg#xywh=100,100,50,50"/>
<img src="kitten.jpg#xywh=pixel:100,100,50,50"/>
<img src="kitten.jpg#xywh=percent:25,25,50,50"/>
```

Figure 4 depicts an example of what this image will look like:

![Figure 4: Spatial dimension example of Media Fragments URI using xywh.js library](image)

The xywh.js library has been created by Thomas Steiner and is provided under the CC0 1.0 Universal (CC0 1.0) license. Finally, another attempt to implement the spatial dimension is also available at http://css-tricks.com/media-fragments-uri-spatial-dimension/. In this case, Fabrice Weinberg shows how some CSS3 techniques can be packaged inside a polyfill and used for the same purposes than xywh.js.

9 http://tomayac.github.io/xywh.js/
2.2.5 Synote Media Fragment Player (SMFP)

The Synote Media Fragment Player enables the automatic creation of semantically annotated YouTube media fragments. A video is first ingested in the Synote system and a new method enables to retrieve its associated subtitles or closed captions. Next, NERD is used to extract named entities from the transcripts which are then temporally aligned with the video. Clicking on every of the results, the corresponding media fragment where the entity has been mentioned is played.

Figure 5: Synote Media Player screenshot

2.2.6 Ligne de Temps

Ligne de Temps is a French project developed by IRI. It is a two-part software: a backend and an interface. The backend enables the user to save his work (notes, editing, rough-cut edits) into a file. It runs the codecs through which videos are imported. The video is encoded and thus readable via the interface. The system generates a timeline out of the movie which organizes the shots and sequences like a score, it can also give access to frequencies and sawtooth patterns. The interface is the visible part of the software through which the user can access and edit content. The interfaces is divided into a menu bar (File, Tool, etc…) and three windows: information, video player and timeline(s). It also supports temporal media fragments without any bandwidth saving. Figure 6 shows a screenshot of the software with a video being annotated; a demo is available at http://ldt.iri.centrepompidou.fr/ldtplatform/ldt/.

Figure 6

---

10 http://www.iri.centrepompidou.fr/outils/lignes-de-temps-2/?lang=en_us
Figure 6: Implementation of Media Fragments URI in LigneDeTemp
3 Media Fragment Creation

This chapter presents the core technologies of Media Mixer in the area of media fragment creation and annotation. In general, video analysis is a well-studied area, and over the years a huge variety of techniques for video content analysis have appeared in the relevant literature. In the chapter, we highlight our implementations of analysis techniques that support three core analysis and annotation functionalities: video shot segmentation, video concept detection, and video object re-detection.

Video shot segmentation is the key enabling technology for automatically generating temporal video fragments, which can then be automatically annotated with concepts and thus become searchable with the use of video concept detection techniques. Object re-detection, on the other hand, is the key to the automatic or semi-automatic creation of meaningful spatio-temporal fragments, corresponding to distinct real-life objects that are depicted in the video.

3.1 Shot Segmentation

3.1.1 Purpose

Video shot segmentation provides the basis for multiple video analysis approaches, such as video semantic analysis, indexing, classification, retrieval, etc. The goal of shot segmentation is to partition the entire video into shorter temporal fragments, namely shots. A video shot is defined as a set of consecutive frames captured without interruption by a single camera. Typically, a video shot contains significant information about specific objects or events that appear in the video, and can be treated as the basic temporal video fragment unit. A detailed review of the state of the art in the area of shot segmentation can be found in [CNI06] and [SJN03].

3.1.2 Material types

The MediaMixer video shot segmentation method is a stand-alone software module. It takes as input a video stream in one of the formats supported by the ffmpeg utility (i.e., avi, mpeg, mp4, etc.).

3.1.3 Methods

The MediaMixer method for the detection of the shot boundaries is based on the algorithm presented in [TMK08]. This approach performs both abrupt and gradual transition detection on the input video stream. In particular, each frame is represented by a color histogram, based on the Macbeth color palette [MMD76], and a color coherence vector [PZM96], which is a two-dimensional color histogram vector that exploits both local and global color information. Moreover, the spatial distribution of pixel intensity is estimated and is expressed by the luminance center of gravity. These three features, i.e. (a) Macbeth Color Histogram Change, (b) Color Coherence Change and (c) Luminance Center of Gravity Change, are then used for detecting both abrupt and gradual transitions. The workflow of segmentation process is presented in Figure 1. The first step of the process is to decompress the video into frames and extract the aforementioned features for each of them. The second step is the computation of the distances between pairs of consecutive and non-consecutive frames based on their feature vectors, forming the corresponding distance vectors. The distance vectors are concatenated into a low-dimensional vector that is used as input to a trained SVM classifier, which decided if a given pair of frames signifies a shot boundary or not. Finally, a flash detector uses the results of the SVM classifier in order to detect any shot boundary classification errors that may have been caused by the presence of flash-lights in the video, and correct them.
3.1.4 Results

The performance of this shot segmentation algorithm was evaluated on various video collections. Here, the results on a News video collection are reported. The ground-truth segmentation was manually generated, leading to 273 ground-truth shots in total.

<table>
<thead>
<tr>
<th>Table 1 Evaluation results of the shot segmentation algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Videos</td>
</tr>
<tr>
<td>Actual Shots</td>
</tr>
<tr>
<td>Correct Detected Shots</td>
</tr>
<tr>
<td>False Positives</td>
</tr>
<tr>
<td>False Negatives</td>
</tr>
<tr>
<td>Precision</td>
</tr>
<tr>
<td>Recall</td>
</tr>
</tbody>
</table>

Both gradual and abrupt transitions occurred in this test video, as well as flash-light camera effects. Precision and recall were used for evaluating the algorithm’s performance. Table 1 presents the results, where it becomes clear that the algorithm performance closes to perfect, which is consistent with other reports on the effectiveness of state-of-the-art shot segmentation techniques. We conclude from these results that the MediaMixer shot segmentation approach can be a reliable and accurate tool for the automatic decomposition of the video into basic temporal fragments.

Figure 8 gives an example of video shot segmentation, where 2 non-consecutive and two consecutive transition frames are presented for a gradual and an abrupt transition, respectively.

Figure 7: Workflow of the shot segmentation algorithm, described in [LinkedTV]
Figure 8: Video shot segmentation example

3.1.5 Constraints

This approach for automatic shot segmentation exhibits some sensitivity in cases of rapid camera movement or changes. In order to avoid the erroneous identification of spurious shots, the algorithm assumes that the minimum allowable duration of each video shot is 25 frames (i.e., one second of video, at 25 frames per second). As a result of this assumption, the algorithm is unable to identify shots that truly last less than 1 second.

3.1.6 Implementation and Licensing

The MediaMixer algorithm, which detects both abrupt and gradual transitions, was designed for both Microsoft Windows and Linux OS. It was implemented in C++ and Matlab. Some parts of the employed method re-use open source software, while the rest is CERTH proprietary software.

3.1.7 Remaining challenges

An important consideration with any kind of visual analysis method, including shot segmentation, is its computational efficiency and thus the speed at which the processing can be carried out. We continue to work on improving this, and as part of this work, we intend to exploit the processing power of the modern Graphic Processing Unit (GPU).

3.2 Video Concept Detection

3.2.1 Purpose

Video concept detection aims to detect high-level semantic information present in media fragments. After the video is decomposed into shots, a concept detection technique is applied to each of these shots so as to extract information about their semantic content. This is an important step for enabling the subsequent organization and retrieval of videos according to their content. A detailed review of the current state of the art in this area can be found in [TREC12] and several publications therein.

3.2.2 Material types

The Mediamixer video concept detection method is a stand-alone software module. It takes a video stream as input, together with the results of video segmentation (coming from, e.g., the method presented in the previous section), so that concept detection can be performed separately for every shot. Trained concept detectors exist for 346 concepts (see [SIN12] for a listing of these concepts). In case one does not want to only use the already trained detectors, but also to train detectors for new concepts, appropriate training data are also needed in the form of ground-truth annotations for these concepts. The supported video formats are the same as those supported by the
shot segmentation method (avi, mpeg, mp4, etc.).

### 3.2.3 Methods

Our method for concept detection is based on the effective combination of a variety of visual features, extracted with the use of multiple shot representations, interest point detectors, descriptors, and assignment techniques, and the use of linear SVM classifiers as described in [MGS+12], so as to correctly associate video shots and candidate concept labels.

The workflow of our system is given in Figure 9. For every shot, sampling is initially performed so that representative key-frames and video tomographs are selected [TA94]. The tomographs are spatio-temporal slices with one axis in time and one in space, representing video motion patterns. So, they are another form of 2D images (though, not necessarily meaningful to humans, as opposed to the traditional key-frames). Two types of tomographs are used in our method, one horizontal and one vertical. After the extraction of key-frames and tomographs, on each such representation of the shot an interest point detector can be applied, so as to sample local image patches. Two interest point detection strategies are used in our method. In the first, image patches are selected through dense sampling (i.e. sampling on a regular grid), while in the second one the detection is performed through a Harris-Laplace corner detector [HS88]. Each resulting interest point is represented using the low-level image descriptors SIFT, RGB-SIFT and Opponent-SIFT, presented in [SGS10].

Subsequently, the extracted low-level descriptors are assigned to visual words using separately two vocabularies that were created off-line through k-means clustering, employing hard-assignment and soft-assignment techniques, respectively [GVSG10]. A pyramidal 3x1 decomposition scheme, employing 3 equally-sized horizontal bands of the image [LSP06], is used for every key-frame or tomograph. So, 3 different Bag-of-Words (BoWs) are formed for each band, while a fourth BoW is built for the entire image. In the end, for each combination of video sampling strategy, interest point detector, descriptor and assignment method, a vector of 4000 dimensions is extracted (this is called a “representation”, denoted in Figure 9, which constitutes the actual input to the utilized SVM classifiers. For classification, linear SVM are employed in order to minimize the computational cost. All classifiers were trained off-line, using the extensive training data that is provided as part of the TRECVID 2012 Semantic Indexing task [OAM+12].

![Figure 9: Workflow of the concept detection technique, described in [MGS+12]](image-url)
3.2.4 Results

The performance of our video concept detection algorithm was evaluated using the video dataset and the concept list that were used in the 2012 TRECVID SIN task, where 46 out of the 346 semantic concepts were evaluated. The detection of these 46 concepts takes place in a video dataset of 8263 videos, fragmented into more than 140,000 shots. For the purpose of the evaluation, the goal of each concept detector was to retrieve the top-2000 shots in which this concept is most likely to appear. The employed detection accuracy measure is the Extended Inferred Average Precision (xinfAP) [YKA08].

An evaluation of three variations of our method conducted, and the experimental results are presented in Figure 10. Based on them, it is clear that the version of the method that employs both key-frames and tomographs achieved higher accuracy for 39 of the 46 concepts and a mean xinfAP value equal to 0.156 (thus increasing the mean xinfAP by 15.5%), compared to the version using only key-frames. More pronounce improvements are generally observed for motion-related concepts (concept labels marked with a * symbol). Based on these results, our proposed configuration for use in MediaMixer applications employs 1 key-frame, 1 horizontal and 1 vertical tomograph per shot. Figure 11 gives a few examples of shot frames and a list of the top 5 detected concepts for each of them.

![Figure 10: Performance of three different concept detection systems](image)

3.2.5 Constraints

This approach for video concept detection uses linear SVMs to minimize the computational cost at the expense of high accuracy. The main drawback is related to the simplicity of the resulting boundary hyper-surface that separates the two classes, which sometimes leads to sub-optimal classification.

3.2.6 Implementation details and licensing

The employed method for concept detection was implemented in C++ and Matlab and it was designed for Microsoft Windows. Some parts of the employed method re-use open source or other publicity available software, while the rest is CERTH proprietary software.
3.2.7 Remaining challenges

The experimental evaluation of the use of video tomographs as an additional sampling strategy for video concept detection showed that the exploitation of motion information is the key for more accurate concept detection. Consequently, research is focused on the improvement of the video tomographs approach and the more elaborate fusion on the different SVMs classifier results.

![Shot Key-frames]

<table>
<thead>
<tr>
<th>Shot Key-frames</th>
<th>Top 5 Detected concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Female person</td>
</tr>
<tr>
<td></td>
<td>Adult</td>
</tr>
<tr>
<td></td>
<td>Door opening</td>
</tr>
<tr>
<td></td>
<td>Doorway</td>
</tr>
<tr>
<td></td>
<td>Indoor</td>
</tr>
</tbody>
</table>

|                | Male person             |
|                | Male human face         |
|                | Indoor                  |
|                | Event                   |
|                | Doorway                 |

|                | Demonstration-protest   |
|                | People marching         |
|                | Walking                 |
|                | Walking-runnn           |
|                | Scene text              |

*Figure 11: Example of video concept detection output*

3.3 Object Re-Detection

3.3.1 Purpose

The goal of object re-detection is to find the occurrences of specific objects within a collection of images or videos. Object re-detection is one of the key technologies for the effective identification and instance-level annotation of temporal and spatio-temporal video fragments that have very similar content. A detailed review of the current state of the art in the area of object re-detection can be found in [PRWG07].

3.3.2 Material types

This object re-detection method is a stand-alone software module. It takes as input a video stream and an image of a specific object of interest, which we want to be re-detected throughout the video. The supported video formats are same as those supported by shot segmentation and concept detection (avi, mpeg, mp4 etc).
3.3.3 Methods

The overall workflow of the MediaMixer object re-detection method is presented in Figure 12. The image provided to the method, showing the object of interest, serves as the query image for the re-detection process. Based on this image, additional images of the object of interest are automatically generated (a zoomed-in and a zoomed-out version of it). Then, the aim of the algorithm is to match any of these versions of the sought object against all video frames, and also localize the detected object at every video frame. To achieve this, it applies feature detection and description both to the query image and all video frames using the SURF algorithm [BETVG08]. Then, each descriptor from the query image is matched against all descriptors from the tested frame, and the best match occurs from a nearest neighbor search (k-NN, for k = 2).

After the matching, a filtering process is applied to clear out the erroneous matches. A SURF key-point in the tested image is kept if the ratio of distances of the nearest neighbors is equal or less than a predefined threshold. In this case, the algorithm keeps the pair of descriptors that correspond to the closest neighbor. The remaining outliers are discarded by applying geometric constraints that estimate the homography between the pair of tested images using the RANSAC algorithm [FB81]. If the matching fails, the algorithm checks the generated versions of the object, firstly using the zoomed-out and then the zoomed-in one. In the end, if one of the three versions of the object of interest is matched successfully with at least one of the examined key-frames then the algorithm continues by matching the object against all the frames of the corresponding shot, using the different versions of the object of interest in the same order as before. Finally, if the object is detected in the video frame, it is demarcated appropriately with a bounding box. In order to achieve high computational efficiency, some of the above processing steps have been accelerated using the GPU.

After the matching, a filtering process is applied to clear out the erroneous matches. A SURF key-point in the tested image is kept if the ratio of distances of the nearest neighbors is equal or less than a predefined threshold. In this case, the algorithm keeps the pair of descriptors that correspond to the closest neighbor. The remaining outliers are discarded by applying geometric constraints that estimate the homography between the pair of tested images using the RANSAC algorithm [FB81]. If the matching fails, the algorithm checks the generated versions of the object, firstly using the zoomed-out and then the zoomed-in one. In the end, if one of the three versions of the object of interest is matched successfully with at least one of the examined key-frames then the algorithm continues by matching the object against all the frames of the corresponding shot, using the different versions of the object of interest in the same order as before. Finally, if the object is detected in the video frame, it is demarcated appropriately with a bounding box. In order to achieve high computational efficiency, some of the above processing steps have been accelerated using the GPU.

Figure 12: Workflow of the object re-detection algorithm

3.3.4 Results

The performance of the algorithm was evaluated using a number of manually selected objects of interest from a set of documentary videos. The experimental results indicate that the algorithm succeeds in detecting the object of interest for various scales and orientations, also in the cases where the object was partially occluded. Figure 13 shows an example of an object of interest and its detection results on a few indicative frames. The developed algorithm leads to unsuccessful detection only in cases when significant changes in scale (extreme zoom-in, extreme zoom-out) and in rotation occur.
Figure 13: **Object of interest (up left) and detected appearances of it.**

### 3.3.5 Constraints

The MediaMixer approach exhibits detection errors in some cases where major changes in scale and rotation occur, and also when the objects of interest have more complicated 3D shapes that are not sufficiently captured by the 2D image that is supplied as input to the algorithm. Also, the size of the object of interest plays a significant role to the algorithm’s performance (with larger, more complex objects being detected with higher accuracy).

### 3.3.6 Implementation details and licensing

The MediaMixer method for concept detection was implemented in C++ and Matlab and it was design for Microsoft Windows. Some parts of the employed method re-use open source software, while the rest is CERTH proprietary software.

### 3.3.7 Remaining challenges

The evaluation of the object re-detection algorithm’s performance showed that this approach achieves quite accurate results. Remaining challenges include the further improvement of the more difficult detection cases mentioned above, i.e. decreasing the algorithm’s sensitivity to significant changes in scale and rotation and to the size of the object of interest. As always, improving the speed of the algorithm is also a challenge.
4 Media Fragment Annotation

Semantic media fragment descriptions permit the connection of self-contained media fragments to the concepts (things, people, locations, events...) they are perceived as representing.

This is important as an extension of usual media description approaches, which consider typically only the whole media item or broad segments, where the segmentation is based on criteria such as specific scenes or shots which may still cover a larger number of concepts. If the media content is to be searchable and retrieved (repurposed, reused) on the other hand at the individual fragment level, then segmentation must be considered at the level of distinct individual concepts, and the media description must both be able to refer to individual fragments and associate them with the distinct concepts they are perceived to represent.

Semantic technology is a means to describe media content in a way which can be understood and processed by machines. Concepts - which can some distinct thing, person, location or event - can be unambiguously identified by URIs using Linked Data principles. By taking the Semantic Web as a basis for the data model and vocabulary, resulting annotations are Web-friendly, i.e. they can be published and retrieved online, additional metadata can be looked up by following URI references, annotations can be manipulated and processed within Web applications. Ontologies – which define permitted terms and how they relate to one another in a formal logical model – are the basis for machine reasoning and automatic derivation of new knowledge about the media, once the media annotation refers to ontological resources such as a Linked Data URI (e.g. a fragment which shows Angela Merkel is also showing the German Chancellor). This article can not consider semantic technology and Linked Data in depth, see the resources section for more.

Semantic descriptions of the media do not exist today directly from media workflow tools, but they can be derived from existing non-semantic metadata generated in the media production process and augmented by tools provided within the media creation phase. The former case is handled by definitions of mappings from legacy metadata formats to the media fragment description format, and the latter is handled by Media Fragment Creation tools.

In this section, we will look at current models and vocabularies used for media resource description. We will consider how to bridge the gap between current approaches and a semantic media fragment description, which requires not only support for the Media Fragment URI specification but also their association to (semantic) concepts, with a particular consideration of the W3C Media Ontology.

Finally, since even this W3C specification proves to not be fully applicable to our needs, we introduce two extensions - the LinkedTV and ConnectME ontologies - which directly address the use case of semantically annotating media fragments for subsequent retrieval and re-purposing in media workflows, and for which supporting tools are in development to help media owners create conformant annotations.

Since those annotations use the RDF data model, any RDF-supporting database can be used to store the annotations and index them for efficient retrieval, using for example the SPARQL query language.

4.1 Existing media description models

In industry usage, media descriptions focus on providing additional metadata about a media item such as its title, description, keywords, date of creation or publication, media-specific characteristics such as dimensions, resolution and duration, and legal information such as owner and usage license. Generally, media description models can be split into two types, the first coming from the digital library domain, and the second coming from the media industry itself. The former is characterized by simple data models (often a set of properties and value restrictions) which can be serialized in various formats (e.g. XML or JSON), and often are media extensions of existing media-independent models. The most typical example would the Dublin Core model, with the complementary Dublin Core Model extensions. On the other hand, to meet the needs of media organizations who desire to capture a much more detailed description of their media content, including low level features extracted by analysis algorithms, segmentation, or the semantic meaning of media segments, the Multimedia Content Description ISO Standard
MPEG-7 is the reference large and comprehensive set of descriptors. In the broadcast industry, the European Broadcaster Union (EBU) has produced several specifications, including TV-Anytime (for EPGs), P-Meta (for archives), EBUCore (for producers) and an extension for the IPTC NewsML-G2 to handle (news) video. For Web media description, lightweight schemas are preferred for exchange between agents, such as Yahoo! MediaRSS or Google Video sitemaps.

The resulting environment is one in which there is a great heterogeneity in the models used to describe media resources, with annotation typically not going into any more depth than general description of the entire media, and the annotation model excludes the (eased) possibility for subsequent application in information interchange with new systems, particularly Web-based [TH07]

From the Semantic Web community, there rose the intention to address the semantic ambiguity of MPEG-7 descriptions (arising from the use of XML Schema as data model, which specifies only syntactic restrictions) which could lead to serious interoperability issues for media processing and exchange [ONH04], [NOH05], [TC04]. The profiles introduced by MPEG-7 and their possible formalization [TBH06] concern, by definition, only a subset of the whole standard. A number of ontologies were proposed, differing in the creation approach, based on MPEG-7 and intended to provide for (near lossless) conversion between XML and RDF representations. The Rhizomik approach (co-developed by MediaMixer partner UdL) consists in mapping XML Schema constructs to OWL constructs following a generic XML Schema to OWL mapping together with an XML to RDF data model conversion [GC05]. COMM, the Core Ontology of MultiMedia [ATS07] has been designed manually by re-engineering completely MPEG-7 according to the intended semantics of the written standard (co-developed by MediaMixer partner EURECOM). The ontologies were designed to allow that the semantic descriptions could be extracted from existing MPEG-7 descriptors (including assumptions about the intended semantics), reusing the segmentation and semantics model of MPEG-7 to allow for regions of media to be connected to semantic concepts. However how the segmentation of the media and its semantics were expressed in the ontology varied from one ontology to the other, resulting in a new heterogeneity, consider the below figure taken from [NDE11] showing the results of 4 ontological approaches (Rhizomik is the top left, COMM is the top right). Furthermore, since MPEG-7 semantics assumed the use of strings to label concepts or possibly a term from a controlled vocabulary, the linking of a semantic concept in the original description to a concept identifier within some global or local concept space would most likely also need to be modeled in the conversion process (numerous online named entity recognition services could provide this, with NERD (http://nerd.eurecom.fr) offering an aggregation service over several, using Linked Data as the most common entity label in a global concept space).

![Figure 14: Comparison of 4 MPEG7 ontological descriptions of the same image region](http://nerd.eurecom.fr)

See [DTKS10] for a state of the art in MPEG-7 ontologies. While this work helped bring the topic of semantic multimedia description to both the Semantic Web and the multimedia communities, it replaced the syntactic complexity of MPEG-7 descriptions with the semantic complexity of MPEG-7 ontological instances, while not addressing non-MPEG-7 media descriptions, growing in mass alongside the general rise of online media content as a whole (e.g. considering HTML META information published with webpage images, or the non-semantic metadata like titles, descriptions and tags associated to social media content on sites like YouTube). This provoked a new effort to tie semantics and Web identifiers into the critical mass of Web-based media descriptions.

4.2 W3C Media Ontology and API

The World Web Consortium (W3C) also considers the lack of interoperability between the different media description models and their limited applicability to the Web as a barrier to the growth of online media (with the appropriate means to search, retrieve, access and re-use when desired by Web based systems). A task force was set up to address this, both in terms of using Web-based approaches (e.g. URLs as identifiers, support for Web data serializations like XML) and within this approach specifying a common vocabulary for media description which could promote interoperability between existing heterogeneous descriptions.

The result of this work is the Ontology for Media Resource, a core vocabulary which covers basic metadata properties to describe media resources (see www.w3.org/TR/mediaont-10/). Since its development took place during other specifications in the Web domain such as the Media Fragments URI, or the use of Linked Data to identify concepts, the W3C recommendation is significant in that it supports these specifications. It is accompanied by an API that provides uniform access to all elements defined by the ontology.

The purpose of the mappings defined in the ontology is to enable different applications to share and reuse metadata represented in heterogeneous metadata formats. For example creator is a common property that is supported in many metadata formats. Therefore, it is defined as one of the properties in the core vocabulary of the ontology for media resources and aligned with other vocabularies. Ideally, the mappings defined in the ontology should be used to reconcile the semantics of a term defined in a particular schema. However, this cannot be easily achieved, due to the many differences in the semantics that are associated with each property in the mapped vocabularies. For example, the property dc:creator from Dublin Core and the property exif:Artist defined in EXIF are both aligned to the property ma:creator. However, the extension of the property in the EXIF vocabulary (i.e., the set of values that the property can have) is more specific than the corresponding set of values that this property can have in Dublin Core. Therefore, mapping back and forth between properties from different schemata, using this ontology as a reference, will induce a certain loss in semantics. The axioms representing the mappings are defined as an exact, broader, or narrower mapping between two properties.

The two MediaOntology properties of most significance to us are keyword and fragment.

- keyword

\[\text{\href{https://www.youtube.com/watch?v=AXgZ98Z9EFw}{\text{Hochwurzen 1850 m.}}}\]

\[\text{596}\]

\textbf{Figure 15: MediaOntology annotation}^{12}

\[\text{\href{https://www.youtube.com/watch?v=AXgZ98Z9EFw}{\text{Hochwurzen 1850 m.}}}\]

\[\text{596}\]

\text{The two MediaOntology properties of most significance to us are keyword and fragment.}

\[\text{\textbullet{} keyword}\]

\[\text{\textbullet{} fragment}\]

\[\text{\textsuperscript{12} courtesy Tobias Bürger and Jean-Pierre Evain, W3C Media Ontology Working Group}\]
A concept, descriptive phrase or keyword that specifies the topic of the resource, using either a URI (recommended best practice) or plain text. In addition, the concept, descriptive phrase, or keyword contained in this element SHOULD be taken from an ontology or a controlled vocabulary.

Note that the keyword SHOULD be taken from an ontology or controlled vocabulary. By specifying URI as an acceptable data type for the keyword value, the Media Ontology makes it clear that Linked Data is a desirable source as a global concept space from which identifiers can be taken.

A tuple containing a fragment identifier and optionally, its role. A fragment is a portion of the resource, as defined by the MediaFragment Working Group.

Note that the fragment identifier is of type URI and is explicitly defined in terms of the MediaFragment specification.

The MediaOntology provides an useful abstraction of the various existing multimedia description vocabularies which can be a basis for metadata interoperability and interchange, but the use of the Media Fragment URI specification or on Linked Data as keywords will only be possible where some equivalent approach (a proprietary segmentation of the media, labeling of media with terms from a controlled vocabulary) can already be found in the original metadata and a wrapper has been specifically prepared to map that approach to the Media Ontology (an examination of the existing mappings at http://www.w3.org/TR/mediaont-10/ notes that keywords and fragments are largely Strings or N/A, as opposed to the preferred use of URIs. With respect to fragments, only MPEG-7 has spatial and temporal decompositions, while EBUCore provides for URI identification of scenes or shots and some other schemas can construct temporal fragments based on start and end times). In other words, the Media Ontology and API can not replace the need for more fine grained (fragmented, semantic) media annotation at the outset, since any mapping must necessarily define explicit knowledge which is only implicit or extracted from the original media (annotation). So, while Media Fragment Creation approaches can work on improving the automated extraction of this knowledge for insertion into a media description, they are best complemented by manual annotation allowing experts to select media fragments and attach concepts to them. The annotation generated in this approach can use the Media Ontology as a basis (e.g. to be able to integrate with further annotations extractable from existing metadata) but can also consider necessary extensions to meet the specific goal of richly associating media fragments with the concepts they are perceived as representing.

From the perspective of implementations, a report was made in 2012 on tests performed on three known implementations of the API for Media Resources 1.0, which supports the use of the Media Ontology for base vocabulary and mappings:13

- Implementation 1: Linked Media Framework of Salzburg Research Forschungsgesellschaft m.b.H.
  The Linked Media Framework (LMF) is a general purpose framework for building Linked Data and Semantic Web applications. It provides commonly used services like Linked Data Server, Linked Data Client, SPARQL Endpoint, Semantic Search, Rule-based Reasoner, Versioning, etc. that can be used for building custom Semantic Web applications using e.g. the LMF Client Libraries available on the project website. The Media-Module builds upon the LMF and provides special services for images and videos (e.g., fragmentation, metadata extraction, etc.).

- Implementation 2: Firefox extension of University of Passau and JOANNEUM RESEARCH
  This showcase utilizes the API in a browser extension following the asynchronous mode of operation. The

13 http://www.w3.org/2008/WebVideo/Annotations/drafts/API/implementation-report.html
application enables a user to generate a video playlist, where videos and corresponding metadata information from different platforms can be arranged in an unified way. As a proof of concept, the browser extension is able to handle YouTube and Vimeo videos.

- Implementation 3: Web service implementation of University of Passau and JOANNEUM RESEARCH
  This implementation is integrated into an image gallery showing images as well as its metadata information. Here, the API is implemented as a Web service following the synchronous mode of operations abstracting from the actual underlying metadata formats. As a proof of concept, Dublin Core and MPEG-7 are used.

4.3 LinkedTV and ConnectME ontologies and annotation tools

Specific use cases generate requirements for a specific ontology and some recent research projects have focused on interactive video, which has a requirement the need to identify specific temporal and spatial parts of a video (media fragments) and to link those parts to other content. In fact, in the case of the projects being introduced here - LinkedTV (http://www.linkedtv.eu) and ConnectME (http://www.connectme.at) - the research goal is to annotate the media fragments with abstract concepts, and let a computer system generate relevant links to content based on the concept (and, in LinkedTV's case, personalized to the viewer's interests). This necessitated the modeling of a specific ontology to capture these requirements.

Both projects have chosen to re-use ontologies as is a common best practice in the knowledge modeling community. The W3C Media Ontology as a media description schema which can interoperate with legacy vocabularies is a natural basis together with the Open Annotation Model which captures information about the actual act of annotating a resource (e.g. who annotated it and when). However, both consider the interactive video use case and extend the base ontologies in two ways:

ConnectME ontology considers it important to also express HOW a concept is represented by a media fragment. It sees the Media Ontology ma:hasKeyword is too weak. We want to capture the form in which the media represents the concept to the media consumer. In order of relationship "strength" this led to the definition of the following subproperties of 'keyword':

- explicitlyShows (in video, image)
- explicitlyMentions (in audio, text)
- implicitlyShows (in video, image is seen something which stands for that concept)
- implicitlyMentions (in audio, text is something which stands for that concept)

The goal of the ConnectME ontology is to allow for media resources to be interlinked based on this richer expression of how concepts are perceived within media. In the guidelines for use of the ontology, it specifies that instances of MediaResource (the atomic media item) are split into MediaFragments and explicitly calls for the use of Media Fragment URIs in the locators of those MediaFragment instances. The use of the richer association of (Linked Data) concepts to media fragments in order to (automatically) generate related sets of media fragments is the subject of ongoing research. The below figure shows a high level view of the ConnectME ontology structure, the ontology is published at http://connectme.at/ontology
LinkedTV ontology (http://data.linkedtv.eu/core/ontology) considers it important to also express HOW content is annotated and interlinked based on media fragmentation and concept extraction procedures. It is a reaction to the tendency to either have very complicated ontologies which are based on direct mappings of the MPEG-7 model (discussed in the previous section) and too simplistic ontologies which can only capture a limited amount of high level characteristics of the media (e.g. the use of the Dublin Core library schema). The W3C media ontology, as a deliberate minimal subset of properties expressed in multiple schema, falls into this latter category. In LinkedTV, the goal is to provide for descriptions which can, at a higher level subset, be interoperable with existing metadata systems (using the W3C Media Ontology) but also are extended with fine grained information on media analysis results and content hyperlinking which is specifically used within the LinkedTV hypervideo workflow. In particular, the purpose of media description in LinkedTV is to express links between spatio-temporal regions of media (audiovisual) content and other Web-based media resources, as input to a hypervideo player. As referenced fully in [LinkedTV2.2], various vocabularies are merged in the ontology to cover the different aspects of media description that are required, including specific means for referencing Media Fragments and expressing their association to (semantic) concepts and (related) content:

- W3C Media Ontology extended with Ninsuna Ontology to explicitly define Media Fragments and their boundaries;
- Open Annotation Ontology for linking media analysis results (concepts) and hyperlinking results (content) to media fragments
- LSCOM, NERD, DBPedia and WordNet as vocabularies and ontologies for unambiguous reference to (semantic) concepts and their types
The below figure represents a high level view of the ontology (from [LinkedTV2.2]), note how media fragments are seen as the base media instance here rather than the media asset they are part of.

**Figure 17: LinkedTV ontology components**

Importantly for an implementation of Media Fragment Description both offer tools which are connected to a media production workflow and can generate valid ontology instances for the annotated media.

In ConnectME, the basis for the media production workflow is the ConnectME Framework. This is a specific extension of the open source Linked Media Framework (LMF) which allows for manual correction and completion of media descriptions via a ConnectME annotation tool and integrates a media interlinking process for hypervideo, in which based on the conceptual annotation of media regions links to related online media are generated in the framework and provided to a dedicated hypervideo player.

With the ConnectME annotation tool, any HTML5-compatible video type can be opened and annotated using an intuitive timeline-based interface. While one can load a video via URL into the tool and annotate it directly, typically a video resource will be 'registered' on a ConnectME Framework instance where some pre-annotation is performed (e.g. currently textual metadata about the video is processed and used to suggest some initial concepts relating to the video, while this could be extended to audio and visual analysis in due course). When a video is loaded from the framework into the annotation tool, this pre-annotation is also provided with the intention that the manual annotation effort can be reduced. Completed annotations are automatically saved into the ConnectME Framework compliant with the ontology, while for convenience the annotation can also be downloaded from the tool in XML or N3 serialization. The tool itself is Web based and can be used in any modern HTML5-supporting browser ([http://annotator.connectme.at](http://annotator.connectme.at)). The below figure shows a screenshot of the annotator with the timeline view of the existing annotations below the video frame:
In LinkedTV, the basis for the media production workflow is the LinkedTV Platform which implements a full media workflow, from media uploading and analysis, through annotation and hyperlinking, to personalization and playout. Various media analysis approaches are integrated into a single tool called EXMERaLDA and the output of this tool is the basis for the annotation phase. Currently LinkedTV has a fully automated process for transforming EXMERaLDA analysis results into a LinkedTV Ontology-based description of the media, with various different services being then called to provide suggestions for links to related content based on the resulting conceptual annotations. In development is an end user tool (called the "editor tool") for visualizing and correcting the automatically generated conceptual annotations and content links, which is seen as a vital prerequisite for LinkedTV technology usage in any commercial workflow. Already, since the LinkedTV use cases are driven by broadcasters, industry requirements are taken into account in the automatic tools, e.g. for content linking a white list of acceptable content sources is provided.

Currently, LinkedTV does provide a 'RDF Metadata Generator' tool which converts media analysis results (ExMERalda files) combined with available video metadata from its source (e.g. the broadcasters own metadata) into LinkedTV ontology compliant descriptions. Currently this is media fragments annotated with semantic concepts. This tool (http://linkedtv.eurecom.fr/metadata/) is still being extended - the next phase will see incorporation of the content hyperlinks into its output, i.e. for every annotated fragment a set of links to related online media will also be provided.

Both ConnectME and LinkedTV are particularly focused on media description for the purposes of supporting hypervideo. Since this requires expressing how specific parts of a video relate to different concepts, and hence can be provided with links to related content, Media Fragment URIs are core to the annotation models. Both ontologies provide a set of properties with which the media fragments can be described, including how they represent particular concepts to a viewer and what online content is in some way further relevant to the content of those media fragments. Tools are under development to allow both automated systems and human users to control and complete media descriptions according to those ontologies. Both extend...
the W3C Media Ontology, which can be seen as the core model for using Media Fragment URIs in media description and attaching some common media properties to those fragments which can interoperate with the use of other media schema across organizations and systems. In the case that a richer fragment description is necessary, the ConnectME ontology provides a minimal extension for a more expressive modeling of 'how' concepts are represented by fragments. The LinkedTV ontology is further extended, allowing that media analysis and interlinking results can also be included and well described, including provenance information.

Hence the selection of which ontology to use in a media description can be based on taking the W3C Media Ontology as basis, and if extensions are necessary, first considering if the ConnectME Ontology is sufficient (more faceted expression of conceptual representation, inclusion of Open Annotation Model to describe the provenance of the conceptual annotation), and then finally looking at the LinkedTV Ontology (inclusion of details about media analysis and hyperlinking results, fuller use of Open Annotation Model to capture not only annotations but also hyperlinking provenance).
5 Media Fragment Management

5.1 Purpose

Media asset management (MAM) addresses the coordination of tasks that manage operations over media lifecycles, such as ingestion, annotation, cataloguing, storage, preservation, discovery, access control, retrieval and distribution of digital video assets. MediaMixer promotes the adaptation of existing media asset management systems to handle media fragments by building a reference system that extends an existing open source media asset repository.

MediaMixer envisages that aspects of MAM are enhanced and adapted to address specific concerns for management of media, creating a consistent digital content object environment capable of handling the fine level of granularity required by the media fragments specification. MediaMixer envisages management of media fragments so that robust, persistent identifiers are maintained to meet needs around both HTTP URIs for web connectivity, as well as industry identifiers. Metadata for web resources addressable by HTTP URIs are an integral part of the semantic web machinery, and MediaMixer envisages that semantic metadata for media fragments are an integral part of robust media asset management, future-proofing media assets for the web.

At the same time, industry identifiers are an integral part of those industry schemes under development that address future rights trading and compliance requirements. MediaMixer envisages that media fragment management utilizes actionable policies with asset management systems that utilize semantic rights metadata, enabling deployment of dedicated policy ontologies that describe permissible conditions of use along the media value chain such as the Copyright Ontology, and the MPEG-21 standards MCO and MCVO ontologies. This would assist automation of access control and compliance checking, and help simplify communication of terms of use to end users.

5.2 Background Drivers and Motivating Scenarios

5.2.1 Emergent economics of preservation and access

The economics of digital preservation are only beginning to be understood. Work undertaken by Presto series of projects has led to more predictable models of economics of preservation and stresses the value of the archive as, in part, a function of usage. The anticipation is therefore that cases will emerge from industry to further strengthen the case for digitisation of those assets currently in non-digital form.

MediaMixer is potentially able to support adding value to the digitisation process through scene detection and annotation, for example, and support for automated metadata enrichment. Digitisation is a (costly) once-only action; a robust, flexible preservation-oriented model will allow maximum reuse of metadata about media fragments identified on ingest to a fine-grained level of detail (whether or not digitisation is performed as part of the ingest procedures).

The well-known, generalised view of preservation-oriented workflows is given by OAIS. The model is by design not concrete in technology terms, as OAIS-based asset repositories are often expected to outlive the lifetime of the

---

14 Notable examples include ISAN (International Standard Audiovisual Number) [http://www.isan.org.uk/](http://www.isan.org.uk/), the EIDR (Entertainment Identifier Registry) [http://eidr.org/](http://eidr.org/), and the LCC (Linked Content Coalition) [www.linkedcontentcoalition.org](http://www.linkedcontentcoalition.org)
16 [http://mpeg.chiariglione.org/standards/mpeg-21/media-contract-ontology](http://mpeg.chiariglione.org/standards/mpeg-21/media-contract-ontology)
17 [http://mpeg.chiariglione.org/standards/mpeg-21/media-value-chain-ontology](http://mpeg.chiariglione.org/standards/mpeg-21/media-value-chain-ontology)
18 Dissemination activities can be found at [https://www.prestocentre.org/4u](https://www.prestocentre.org/4u)
technology system that manages them at any one time.

**Figure 19: OAIS Functional Model**

### 5.2.2 Heterogeneity in tooling and delivery networks: towards cloud-based solutions

Channel operators within the media value chain are brand owners acquiring the (rights to use) content for the purposes of playout. Channels may be destined for Internet streaming, high quality TV viewing through a home appliance, or for carriage on cable or satellite. In all cases, asset management workflows should be coherent from ingest to distribution. Operators’ asset management needs typically cover workflows from ingest to transcoding, assignment of metadata, scheduling, management and hand-off or CDN distribution. Operators are able to free up capital and resources by allowing management of these various aspects of creation and operation of a linear channel in the cloud (public, private, or hybrid). Other advantages include elimination of equipment compatibility and long-term technology trade-offs, allowing potentially faster time to distribution. With asset management deployed as a cloud-based service, capital budgets could for example, be re-focused on content, audience development and monetization of viewership.

MediaMixer permits fine-grained reuse of content within channels due to addressing audio-visual materials as fragments in URI form, which provides the basis for addressability on the web, and inherently cloud-based. Furthermore, web-based information enrichment within such channels becomes possible due to web ontology standardization, protecting investment in workflow outputs such as semantically enriched and annotated ingested material, for future connected TV, second-screen applications etc. Information enrichment and annotation is, for example, potentially valuable for brand owners for advertising purposes.

MediaMixer envisages deploying its media asset management technology on the open source cloud software OpenStack, maximizing portability across deployment models and networks.

---

http://www.openstack.org/
5.2.3 From files to digital objects

One of the primary challenges to enable fragment management and reuse is to address linkage and interoperability between the metadata and content artefacts created by ingest processes. Approaches on ingest will vary: some ingest procedures may rely purely on cataloguing and metadata production, without any persistent preservation requirement for digitized content at all. A robust digital object model stresses the separation of concerns between the intellectual resource, with which certain descriptive metadata may be associated, and its representational forms, with which technical metadata may be associated. Allocation of identifiers to digital objects allows portability and independence from any one physical filesystem, and allows them to be interrelated.

The representation of all metadata in semantic form allows a knowledge base to be constructed and managed as an integral part of media asset management. The representation of knowledge conforming to W3C semantic web standards allows enrichment of the knowledge base with information from the cloud of linked data, dissemination of metadata as linked data resources themselves, and the B2B exchange of information between parties participating within the media value chain.

5.2.4 Collection management and archive management

Access, use and re-use is noted as being inextricably linked to the value of an archival collection. Media asset management for MediaMixer treats knowledge about assets controlled by a party as a “first class citizen”. Knowledge is being amplified currently in commercial spheres outside the domain of audiovisual materials due to the explosion of interest in big data, analytics, linked open data and the semantic web. MediaMixer asset management is able to capitalize on these industry trends through its emphasis on the use of web standards for knowledge representation. MediaMixer supports the provision of the three fundamental archival services, support for discovery, access, and sustainability.

Creating a single consolidated source of information on which to base business processes is a primary function of any media asset management system. Semantic enrichment of the assets' metadata in an archive increases the value of the curated assets. With an increase in findability and fine-grained access to material for production purposes (and other factors such as rights being equal), increases in subsequent reuse will have a positive impact on the economics of production overall.

The architecture of the media asset management environment envisioned by MediaMixer is one that connects MediaMixer components with content they process, and incorporates them within an open extensible architecture to maximize the economics of preservation and reuse potential.

Collection management can be an important aspect of media asset management. Well-managed collections often embody deep and sought-after knowledge about the curated assets. From the collection management, archival and preservation points of view, the ability to plan budgets, timelines, equipment needs, and other preservation plans that unequivocally impact access is directly tied to the documentation of some degree of item-level knowledge about one’s collection. Within MediaMixer, item-level knowledge is, potentially, at the very fine-grained, media fragment level. To participate effectively within an anticipated “web of media fragments”, media asset management technology must therefore be adapted to include fragments as an integral part of the ingest and cataloguing process.

---

21 Recent analysis available at https://www.prestocentre.org/library/resources/assessing-audiovisual-archive-market
22 For example, as identified by AV Preserve: http://www.avpreserve.com/wp-content/uploads/2012/08/WhatsYourProduct.pdf
5.3 Material Types and Methods

5.3.1 Digital asset management system based on Fedora Commons

Fedora Commons\(^{23}\) is an open source digital object repository, licensed under the Apache 2.0 license, often regarded as a digital asset management system, for all types of media. Fedora Commons supports the OAIS model. The services used for MediaMixer, as integrated with Fedora, focus on Fedora’s abilities to handle video as the primary digital content type (although Fedora can be used as a metadata-only records environment: media does not necessarily require digitization to maintain a catalogue of holdings that may be digitized in future, or “on request’ and processed for fragment detection, annotation, etc).

Fedora is used for storing, managing, preserving and accessing digital content using a quite abstract notion of a digital object. The flexibility this provides presents a challenge to rapid deployment needs, but also affords an opportunity to adapt Fedora’s model to semantic techniques where valuable to do so.

As Fedora is a popular system with open source users, and especially in the institutionally-funded sector, rapid deployment needs can to some extent be satisfied with the also-popular open source frameworks used for video materials, such as Islandora\(^ {24}\) and the Avalon system\(^ {25}\). These environments configure Fedora for use in Drupal\(^ {26}\) and Ruby\(^ {27}\) web application stacks, and significantly lessen the time to deployment to meet basic needs of any media collection, as they provide common workflows and application services.

5.3.2 Identifier Framework

Fedora is more a framework than an out-of-the-box content management solution. Fedora adopters tend to expect to invest significant time in repository design and configuration prior to going into production. This is reflected in its typical role as a system capable of meeting preservation demands, and situations that demand that identifiers for content, or specific versions of content, reliably resolve to exactly the content expected. Given there are the appropriate organizational procedures in place, this often lends itself to situations demanding “persistent identifiers” - durable identifiers suitable for associating with embedding into (digital and non-digital) video materials.

Fedora arranges content according to a flexible object model, each node of which is allocated an internal PID (Fedora “persistent identifier”) that uniquely resolves to the node. Fedora PIDs are under the control of the owner of the Fedora-based media asset system. Fedora may also be used with externally-controlled persistent identifier systems, such as the handle\(^ {28}\) or DOI system\(^ {29}\), to preserve guarantees of persistence with content agreements that rely on these mechanisms. One such emergent DOI-based system relevant to MediaMixer is the EIDR. Allocating and distributing DOIs for managed content, for example, is one route to publication of metadata associated with the media entity on the web, as information about all DOIs are available through gateways\(^ {30}\).

Natively, Fedora is able to present identifiers using URIs to the web using its native APIs, based on REST and SOAP, and, optionally, a SPARQL interface to its RDF-based store of typed object-object relationships, the Resource Index.

\(^{23}\) [http://fedora-commons.org/](http://fedora-commons.org/)
\(^{24}\) [http://www.islandora.ca/](http://www.islandora.ca/)
\(^{25}\) [http://www.avalonmediasystem.org](http://www.avalonmediasystem.org)
\(^{26}\) [http://drupal.org/](http://drupal.org/)
\(^{27}\) [http://projecthydra.org/](http://projecthydra.org/)
\(^{28}\) [http://www.handle.net/](http://www.handle.net/)
\(^{29}\) [http://www.doi.org/](http://www.doi.org/)
\(^{30}\) For example, [http://dx.doi.org/](http://dx.doi.org/)
5.3.3 Digital Object Model

In many ways the Fedora repository is similar to a web CMS in its role of storing and providing access to digital content, but with a heavy focus on preservation and a flexible content model. Unlike the more usual CMS hierarchical content models, Fedora's objects are structured as a graph of content nodes.

![Fedora Digital Object](image)

**Figure 20: Fedora Digital Object**

Fedora provides a specific kind of "resource-oriented" view of a networked and potentially very large repository of content. Contents can be accessed and transformed via services beneath that resource-oriented model. Contents themselves can be textual, audio-visual, or indeed any bitstream content. The "repository" can be viewed as essentially middleware managing contents that can be physically distributed.

Fedora is very flexible in its possibilities and not very prescriptive about the structural arrangement of objects. Fedora objects are typically a compound aggregation of one or more closely-related content items (datastreams). Datastreams can be of any format, and can be either stored locally within the repository, or stored externally and referenced by the digital object.

The Fedora model makes explicit the difference between a conceptual resource ("the object") and its bitstream "representation" via datastreams (as indicated above, these are however not the same thing as "representations" in the web architecture sense\(^{31}\)). An instance of a datastream can be thought of as a manifestation (serialisation) of some digital object, or a manifestation of some metadata about the object. For example, a mobile and digital master versions of a video would typically be arranged as datastreams of "the same" digital object, which is conceptual in nature. A metadata record in a certain format or structure (like XML) about the image would also be typically treated as a datastream attached to it.

Fedora's graph of content includes relationships between nodes representing the conceptual digital objects themselves, between objects and datastreams, and triples expressing properties of both objects and datastreams. Digital objects can represent any object, a capability exploited in the case of representing license agreements for content in MediaMixer themselves as objects, linked to "policy objects".

---

\(^{31}\) For further details see Appendix B of RIDIR Report available at: [https://edocs.hull.ac.uk/muradora/objectView.action?pid=hull:1003](https://edocs.hull.ac.uk/muradora/objectView.action?pid=hull:1003)
Advantages in creating objects for significant entities is two-fold:
- Objects may be serialized, for content migration, upgrade, long-term preservation and disaster recovery
- Objects may be indexed within Fedora’s semantic store

5.3.4 Content Model Architecture (CMA) and Behaviours

Objects can be "typed" by specifying a "profile" object that defines the pattern of datastreams expected for objects expressing homogeneous content. These objects are known as Content Model objects.

Fedora takes a “resource oriented approach” in that services, or “behaviours”, are attached to identified digital objects. Services are attached using conceptual links, and an implementation provided, typically using a REST- or SOAP-based service, akin to the object-oriented programming notion of interface and implementation. The approach lends itself well to integrations with external management of web resources that use a resource-oriented (REST-based) programming model.

Services may be included as part of Content Models. In conjunction with Fedora’s semantic capabilities, this leads to a robust and powerful preservation-oriented programming model for local and remote services to manage “web resources”, those entities identified by URI, in a way that is neutral to the resource’s physical location. For example, a datastream can be used to specify a certain type of video format in a content model, one that when accessed resolved to content being sourced from a different format and transcoded on the fly.

This may be of potential use to applications working with MediaMixer capabilities, as it allows for “profiles” of content to be created within an existing archive, based on automated content discovery procedures; for instance, to support cataloguing applications.

5.3.5 Semantic Web Capabilities

As mentioned, Fedora has native semantic capabilities, being bundled with the open source Mulgara RDF triple store32.

An additional method, which is being employed for MediaMixer, builds upon the integration mechanism developed as part of the IKS Project33. Updates to Fedora Commons objects are reflected in JMS34 messages to a component attached to a semantic knowledge store, thus causing the store to become the Resource Index. This integration becomes a powerful combination when the semantic services offered with the knowledge store go beyond those offered with a basic RDF triple store, such as in the case of Apache Stanbol, integrated for IKS.

The integration allows Fedora object metadata held within the media asset repository to be interfaced to the web in a more sophisticated way, for example, to service linked data protocols. The separation of the concerns of the web architecture is important since the W3C and the web architecture controls key notions such as “web resource”, which are unlikely to be semantically equivalent in all situations to those “resources” under media asset control. The bitstream associated with a (URI-identified) web resource can be the subject of content negotiation, a function between the web agent machinery (web browser or linked data agent) and the resource provider (web server), whereas for preservation or other object identification purposes this may not be desirable, and an identifier may be a token, strictly valid for a single bitstream representation alone.

Construction of more sophisticated reasoning, inference, knowledge-integration and support for more advanced classification and cataloguing services will also provide additional value from a dedicated semantic services

---

32 http://www.mulgara.org/
33 http://www.iks-project.eu/
34 For overview, see http://en.wikipedia.org/wiki/Java_Message_Service
capability integrated with Fedora Commons. Knowledge integration can for example include association of existing SKOS-based thesauri used to catalogue assets with the descriptive metadata associated with the assets, which may also then be automatically, via inference, used with the Media Ontology annotation relationships. This can for example provide keywords from a controlled vocabulary within a linked data set, published onto the web to aid discovery of assets within a media collection and drive traffic to the collection.  

A further use of semantic services is to define SOLR indices to support highly performant keyword-based and faceted searches, built purely from an integrated semantic knowledge base about the assets.

5.3.6 Content Protection Measures

Licensed content held within an archive must be managed effectively, according to policies that reflect license agreements made between parties. Licenses should determine under what conditions which parties may perform which actions to which resources. Asset bases must respect such agreements.

Media asset management for MediaMixer assists the media manager responsible by implementing an effective access control mechanism, built using the Fedora Enhanced Security Layer (FeSL), that is capable of storing license agreements (as documents) as an integral part of the archive. Agreements are then converted to machine readable access control documents, expressed using the XACML security policy expression language. The XACML expressions are written such that attributes contained within its constituent rules may be sourced from the knowledge base (RDF-based semantic graph).

The link between original license documents and machine-enforceable rules is preserved as part of the archives overall integrity, (termed “policy objects”), with semantic linkage to link to those elements that are held within the asset store’s knowledge base, referenced as URIs. The knowledge base may legally include fragment URIs, allowing policies to be built that integrate with specific fragment types, for example. An open XACML Policy Decision Point (PDP) API allows other integration points, such as with an LDAP store to enable hierarchical role-based user access control to be used in conjunction with semantically-grouped resources.

5.3.7 Support for rights

Sophisticated reasoning capabilities are required, for example, when evaluating licenses, expressed using the Copyright Ontology. To maintain trustworthy status, license agreements in machine-readable and machine-interpreted form must be curated to the same high standards of integrity as other valuable assets. To correctly interpret and evaluate license agreements a burden is placed on the media management environment to not only curate asset metadata to item-level, but to handle identifiers of those entities (agentive parties and resources) specified within the agreement, and maintain an appropriate audit trail for reporting and exchange purposes. Only with reliable management of the input data can inferences and calculations made about license agreements also be considered reliable. Entity management facilities that allow multiple URIs to be associated with resources, are included with the Stanbol framework.

5.3.8 Provenance and Trust

Fedora is able to record changes natively within an “AUDIT” datastream.

In addition, previous work has been undertaken to integrate provenance ontology with Fedora’s semantic capabilities, to build a queryable semantic knowledge base to record the actions taken over the asset repository’s

35 For previous work, see https://conferences.tdl.org/or/OR2011/OR2011main/paper/view/470/109
36 http://lucene.apache.org/solr/
37 https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml
38 For description, see https://conferences.tdl.org/or/OR2011/OR2011main/paper/view/473/149
resources.

5.3.9 High Level Storage API

An abstraction at the storage API level allows for flexible storage options, using a pluggable mechanism that does not impact the digital content representation of objects in the asset repository. Options include cloud-based storage\(^{39}\) as well as physical Akubra-based\(^{40}\) storage.

“Hints” are provided to the underlying storage implementations to enable optimisation.

Fixity checks etc. are implemented to protect against generational loss and help preserve integrity.

5.4 Constraints

5.4.1 Facilities outside core MediaMixer concerns

Storage media associated with MediaMixer are not envisaged to go beyond the facilities offered by Fedora Commons.

The video transcoding facilities are not envisaged to extent beyond those already offered by the media asset management frameworks built using Fedora, which are primarily based around FFmpeg\(^{41}\). However, the method is easily extensible to incorporate other facilities.

5.4.2 Summary list of Fedora Commons features

- Digital content of any type can be associated, stored, managed and maintained, not only video materials
- Metadata about content in any format can be managed and maintained
- Scales to millions of objects
- Access data via Web APIs (REST/SOAP)
- Provides RDF search natively (SPARQL)
- Enhanced semantic services, via JMS to Apache Stanbol (or similar framework)
- Disaster recovery and data migration support via the Rebuilder Utility
- Repository rebuild facility from digital object and content files for robust preservation
- Content Model Architecture defines "types" of objects by their content
- Multiple storage options (database and file system abstractions, allows e.g. flexibility for indexing and tape options)
- JMS messaging (integration method whereby applications and services can "listen" to repository events)
- Web-based Administrator GUI for low-level object editing support
- OAI-PMH Provider Service for metadata exchange
- Fulltext Search Service using GSearch
- Enhanced Security Layer for declaratively specifying Access Control policies

5.5 Remaining Challenges

The policy object mechanism that encapsulates XACML expressions does not currently have a user-friendly editor, or an offline policy evaluation feature to test policies. In addition, sourcing XACML attributes from the Resource

\(^{39}\) [http://www.duracloud.org/](http://www.duracloud.org/)
\(^{40}\) [https://wiki.duraspace.org/display/AKUBRA/Akubra+Project](https://wiki.duraspace.org/display/AKUBRA/Akubra+Project)
\(^{41}\) [http://www.ffmpeg.org/](http://www.ffmpeg.org/)
Index RDF may in some circumstances lead to issues with dynamic behavior. Options in each case are currently under investigation.

The integration with Apache Stanbol is not yet to production standard, and integration with Stanbol’s RDF triple store, based on the Apache Clerezza RDF triple store API, has only been tested for use with the Stanbol OntoNet ontology management features. Its use with other semantic capabilities of Stanbol has not been fully tested.

Integrated support for reasoning and inference for rights management purposes has not yet been implemented. Correct implementation of Fedora content models to support media fragment URIs, with part/whole semantics. There may be challenges to reconcile W3C media fragments with other fragment identifiers that may arise, for example, a DOI application to use the handle system to manage fragments as extensions (handle system 7.04, which omits the requirement to mint a new handle per fragment).

A remaining challenge will be to produce media resource descriptions using the Ontology for Media Resource. This can build on previous work, where the OntoNet component of Apache Stanbol has been used to align heterogeneous ontologies, link and merge RDF data. In particular, taxonomic thesauri were aligned with the Fedora Commons ontology resources that describe relationships between Fedora object model entities.

OntoNet’s sessions and scopes feature allows the partitioning of ontologies, independent from the W3C OWL imports mechanism, which in some circumstances can be unsatisfactory for managing the operational characteristics of inference and processing rules. An organizational structure of ontologies is to be determined by specific use cases and inferences required.
6 Media Fragments Rights Descriptions and Negotiations

This chapter describes copyright management based on the Copyright Ontology and its implementation based on the Web Ontology Language (OWL).

6.1 Purpose

Digitalization and the transition to a Web full of media, where video already amounts more than half of online consumer traffic\(^{42}\), have introduced new scalability requirements like bandwidth exigencies, which technology is rapidly evolving to cope with. However, there are other limiting factors that are not scaling so well, especially those that have been traditionally slow moving like copyright.

As the amount of content made available through the Web grows, for instance 72 hours of video are uploaded to YouTube every minute\(^{43}\), the problem of managing its copyright becomes even more relevant. Consequently, there is already a need to make rights management scale to a web of media, as pointed by recent initiatives like the Picture Licensing Universal System\(^{44}\) or the Linked Content Coalition\(^{45}\). These initiatives, among others, propose ways to represent and communicate rights so they can be automatically processed in a scalable way.

However, the issues associated with copyright management at a Web scale become even more complex when it goes beyond simple access control and takes into account also content reuse and the whole content value chain. In this case, rights representations need to be more sophisticated so they can capture the full copyright spectrum.

In addition, as reuse is easier when considering just fragments, spatial or temporal, of existing content and not full content pieces. Proposed solutions should scale not just to a Web of media but also to a Web of media fragments. Fragments, accompanied by scalable copyright management for the full value chain, enable a potentially enormous re-use market.

The Copyright Ontology is implemented as a Web ontology that facilitates the representation and communication of rights and licensing terms over media assets in terms of their fragments. The ontology is based on Semantic Web technologies and integrates with the W3C Media Fragments Recommendation\(^{46}\) to define and describe spatial and temporal media fragments.

The ontology makes it possible to underpin the media discovery and usage negotiation process, facilitating the automation of functionalities for rights management. Based on an explicit and interoperable semantic representation for the communication of rights, the ontology facilitates assessing the reusability of a given media asset fragment and eases bringing content onto this flourishing market. For instance, by interoperating with DDEX data,\(^{47}\) one of the main standards for automating the exchange of information along the digital supply chain.

6.2 Scenario motivating copyright management of media fragments

A media production company wants to check if the copyright of media sources it is re-using are well aligned with internal policies and the special agreements it has with some of these media providers and the rights holders. However, it is not feasible to manually check the rights of each individual media source, many of them media

---


\(^{44}\) PLUS, http://www.useplus.com

\(^{45}\) Linked Content Coalition, http://www.linkedcontentcoalition.org


\(^{47}\) DDEC, http://www.ddex.net
fragments. Agreements and policies are document-based while media sources are accompanied by DDEX and other rights expression languages. The former is a relatively small set of documents that can be formalized manually (with some assistance from an editor tool) using semantic technologies and the Copyright Ontology. The rights data for media sources is automatically mapped to the same semantic model so it is possible to perform automated reasoning to check if there are conflicts among media sources terms and the policies and agreements.

Fragment detection and semantic annotation of media fragments allows finer-grained and more informed agreement and policy checking. For instance, the policy includes avoiding violent media fragments when producing content for children. Incoming assets are processes to detect fragments including violent scenes (war, weapons, explosions,...) so the reasoner can take this information into account to automatically detect this kind of conflicts.

6.3 Material Types

The ontology takes into account the different forms a creation can take along its life cycle:

- **Abstract**: Work.
- **Objects**: Manifestation, Fixation and Instance.
- **Processes**: Performance and Communication.

The ontology also includes some relations among them and a set of constraints on how they are interrelated as shown in Figure 21.

![Figure 21: The creation model provided by the Copyright Ontology](image)

6.4 Methods

The ontology is implemented using the Web Ontology Language (OWL). Different reasoners can be then used to provide:

- **Consistency checking**: detect if a set of licenses is consistent and thus it is authorizing a set of actions that is not empty.
- **License checking**: based on the subsumption service provided by the reasoners it is possible to detect how licenses interact, for instance detecting licenses that completely include other licenses making them not necessary. It is also possible to perform license search based on example
licenses, so it is possible to detect if there is a license that would provide the functionality of a fictitious one.

- **Usage checking**: based on the reasoned instance classification service to detect if a particular action, for instance copying a media fragment, is authorized by a set of licenses. This feature is based on the ability of reasoners to check if the action satisfies all the restrictions set by a license. For more details about this feature see\(^\text{48}\).

## 6.5 Results

The Copyright Ontology has been applied in real use cases, for instance involving DDEXT rights data. DDEXT data is used in this case as the way to communicate the rights associated to assets along the value chain. However, DDEXT data does just model deals, which capture the kind of actions that can be performed with a particular asset or fragment in a given territory, time point, etc. They do not capture the existing copyright agreements that might make those particular actions legal or not. Table 2 includes a DDEXT example on the left column.

Consequently, if there is a dispute because an asset or fragment is detected under a conflicting use, it is difficult to determine if there is legal support to claim compensation. Many different DDEXT deals might be involved and even the agreements related with the involved assets might have to be manually checked. This is not feasible if the amount of disputes to deal with grows.

<table>
<thead>
<tr>
<th>&lt;Deal&gt;</th>
<th>&lt;DealTerms&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;DealTerms&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;CommercialModelType&gt;PayAsYouGoModel</td>
<td></td>
</tr>
<tr>
<td>&lt;CommercialModelType&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;Usage&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;UseType&gt;OnDemandStream&lt;/UseType&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;DistributionChannelType&gt;Internet</td>
<td></td>
</tr>
<tr>
<td>&lt;DistributionChannelType&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;TerritoryCode&gt;ES&lt;/TerritoryCode&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;TerritoryCode&gt;US&lt;/TerritoryCode&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;ValidityPeriod&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;StartDate&gt;2013-01-01&lt;/StartDate&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;/ValidityPeriod&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;/DealTerms&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;/Deal&gt;</td>
<td></td>
</tr>
</tbody>
</table>

| <http://media.com/deals/3> | owl:Class, msp:Deal; |
| co:start "2013-01-01" ; |
| co:alm ddex:PayAsYouGoModel ; |
| owl:intersectionOf { |
| ddex:OnDemandStream |
| [ a owl:Restriction ; |
| owl:onProperty co:theme |
| ; |
| } |
| owl:hasValue |
| <http://my.tv/video.ogv#t=60,100> } |
| [ a owl:Restriction ; |
| owl:onProperty co:medium ; |
| owl:someValuesFrom |
| ddex:Internet ] |
| [ a owl:Restriction ; |
| owl:onProperty co:location ; |
| owl:someValuesFrom |
| [ a owl:Class ; |
| owl:oneOf |
| (territory:ES territory:US) ] |
| ]). |

### Table 2: DDEXT data example (on the left) and the corresponding model based on the Copyright Ontology (on the right)

DDEXT has been mapped to the Copyright Ontology, so DDEXT data can be converted into Semantic Web data based on this ontology. This way, many different deals can be combined and taken into account to decide a dispute. Moreover, they can be also combined with other sources of information, like existing agreements once they are also formalized.

Once combined, it is possible to use reasoners to easily implement the process of checking if the dispute being considered is supported by any of the existing deals or agreements. To do that, deals are modeled as classes based on the intersection or union of restrictions on the deal action and its case roles, as shown on the right of Table 2. These classes define the set of actions that are authorized by a deal. The reasoner can then be used to check if the dispute, modeled as an instance, is inside the set defined by the class and consequently it can be interpreted as

---

\(^{48}\) Copyright Reasoning Explained, http://community.mediamixer.eu/materials/presentations/copyright/view
supported by the deals and agreement under consideration, as illustrated in Figure.

Figure 22: Illustration of reasoner classification service for checking if copyright dispute is supported by existing license deals

6.6 Alternative tools

The DRM Watch review on DRM standards\(^{49}\) shows that interoperability is a key issue for DRM systems. For instance, it arises in the content distribution scenario when a users want to consume content in any of the devices they own. Interoperability is also critical in the organization scenario, when content flows through organizations or external content is used in order to derive new one.

The main response to DRM interoperability requirements has been the settlement of many standardization efforts. The main ones are ISO/IEC MPEG-21\(^{50}\) and ODRL\(^{51}\), and in both cases the main interoperability facilitation component is a Rights Expression Language (REL).

The REL is a XML Schema that defines the grammar of a license modeling language, so it is based on a syntax formalization approach. There is also the MPEG-21 Rights Data Dictionary and a ODRL Data Dictionary Schema (DD) that captures the semantics of the terms employed in the REL, but it does so without defining formal semantics\(^{52}\).

This syntax-based approach is also common to other DRM interoperability efforts and one of main causes of the proliferation of interoperability initiatives that cannot interoperate among them, like in the e-books domain\(^{53}\). Despite the great efforts in place, the complexity of the copyright domain makes it very difficult to produce and maintain implementations based on this approach.

The implementers must build them from specifications that just formalize the grammar of the language and force the interpretation and manual implementation of the underlying semantics. This has been feasible for less complex

---


domains, for instance when implementing a MPEG-4 player from the corresponding specification. However, this is hardly affordable for a more complex and open domain like copyright, which also requires a great degree of flexibility.

Moreover, the limited expressivity of the technical solutions currently employed makes it very difficult to accommodate copyright law into DRM systems. Consequently, DRM standards tend to follow the traditional access control approach. They concentrate their efforts in the last copyright value chain step, content consumption, and provide limited support for the other steps.

In fact, just Internet publishing risks are considered and the response is to look for more restrictive and secure mechanism to avoid access control circumvention. This makes DRM even less flexible because it ties implementations to proprietary and closed hardware and software security mechanisms.

The limited support for copyright law is also a concern for users and has been criticized, for instance by the Electronic Frontier Foundation. The consequence of this lack is basically that DRM systems fail to accommodate rights reserved to the public under national copyright regimes.

Consequently, the DRM world remains apart from the underlying copyright legal framework. As it has been noted, this is a risk because DRM systems might then incur then into confusing legal situations. Moreover, it is also a lost opportunity because, from our point of view, ignoring copyright law is also ignoring a mechanism to achieve interoperability. Therefore, DRM must evolve to Copyright Management.

It is true that copyright law diverges depending on local regimes but, as the World Intellectual Property Organization promotes, there is a common legal base and fruitful efforts towards a greater level of copyright law worldwide harmonization.

A new approach is necessary if we want profit from the Internet as a content sharing medium. The existence of this opportunity is clear when we observe the success of the Creative Commons initiative, whose objective is to promote content sharing and reuse thorough innovative copyright and licensing schemes.

However, despite the success of Creative Commons licenses, this initiative is not seen as an alternative to DRM. The main reason is the lack of flexibility of the available licensing terms. There are mainly six different Creative Commons licenses, all of them non-commercial, and just an informal mechanism for extension and adoption of alternative licensing schemes, CC+.

6.7 OS/costs

The Copyright Ontology is available under a Creative Commons license that just requires attribution and allows commercial uses.

6.8 Remaining Challenges

The main remaining challenge is basically related with the current state of the art of scalable databases with reasoning capabilities that can load licenses based on the Copyright Ontology and provide enough reasoning capabilities when the amount of licenses and clients scales. However, latest results with reasoners like Ontotext

57 http://wiki.creativecommons.org/CCPlus
OWLIM\textsuperscript{58} have shown high performance and linear scalability through replication clusters.

\textsuperscript{58} Ontotext OWLIM, http://www.ontotext.com/owlim
7 Media Fragment Lifecycle

The following chapter describes the media fragment lifecycle from creation up to consumption and by this also integrates the different aspects as described in the previous chapters. As the core technologies are more deeply covered in these chapters this chapter focuses on the description of the overall lifecycle, noting additional technologies wherever appropriate.

The whole media fragment lifecycle can be roughly divided into three main parts (cf. Figure 23): a) video analysis, b) metadata aggregation and c) the publishing and consumption part. These different lifecycle stages have to be embedded into a production environment, including the technical infrastructure, the processes, the different stakeholders including producers, license holders, publishers, editors, re-using parties, consumers, etc.

Figure 23: Overview of the Media Fragment Lifecycle

In the following these different stages are described in more detail.

7.1 Analysis

The purpose of the analysis workflow is to create media fragments out of original media resources. The main steps here are 1) ingestion, 2) the video analysis itself and 3) the generation of media fragments.

Figure 24: Overview of the Media Resource Analysis process

7.1.1 Media Resource Ingestion

The first step is here to select and import the media resources which shall be made available for later media fragment re-use. This step is the media resource ingestion. The selection of media resources here depends on the productive context: in case of TV broadcasters or similar producers the ingested content is mainly the content produced by the station itself. Here it can be integrated into the planning and production workflow. Besides the facts that the material to be analysed exists in the highest quality and that the analysis process can be planned in advance, the most eminent advantage is that all metadata which is generated during the production process is already available. Although most of the metadata, such as title, genre, cast, description, technical formats, etc. applies to the media resource as a whole rather than to fragments of it, some metadata information can already by used, in particular the subtitle information and, especially in news shows, the information concerning single reports. A further advantage is, that the producers are in full control of the Digital Rights Management, licensing and billing rules.

---

59 The following chapter to a great extent summarizes results from the EU Project LinkedTV [LinkedTV]. For further details refer to the public Deliverables [LinkedTVD5.1] and [LinkedTVD5.3].
This is more complicated in other business scenarios where some provider operates over-the-top services offering media fragment generation and re-use after publishing. Here the range can reach between direct contracts between producers and media fragment providers unto the ingestion of completely open material available over the internet via video portals or streaming services. While in the former case through the service contracts additional metadata might be available, only metadata is available which is either exposed or extractable from that portal. In the case of streamed material the ingestion of course only makes sense if the stream can be recorded and stored for analysis and later publishing. Special connectors have to be provided for each source which is selected for ingestion.

The main steps within the ingestion process are:

1. Selection: the selection of the source from where media resources shall be imported, like TV production system or Internet Video Portal, etc. This might include subscribing to special channels and observing
2. Import media resource: in case the analysis cannot per formed remotely or at the place where the original content resides, the media resources have to be stored at a local media repository
3. Import metadata: import all related metadata which is provided by the publisher (like subtitle information, TV-Anytime program information, etc.) via API or extract from the related website.
4. Publish metadata: in order to make the media resource available for video analysis the URI and related metadata has to be exposed via an interface, e.g. as a REST service; the media analysis components may subscribe to a respective event notification and thus the subsequent process can be triggered.

7.1.2 Media Resource Analysis

After the completed ingestion the video is being analyzed through the media resource analysis process. This process includes a collection of different subprocesses each specialized in analyzing different aspects, like detecting segments or shots, analyzing the audio track, detection of special visual concepts or objects, faces, moods, etc. For a deeper description of the media resource analysis step refer to Chapter 3 Media Fragment Creation.

The media resource analysis process is – very much like the text processing for textual search engines – a multimedia indexing process, i.e. it results in an index for each analyzed video, at which temporal and/or spatial segment detected objects, concepts, etc. occur, although the resulting index in itself is again textual and not visual or audio-based.

The media resource analysis is generally open for 3rd party tools which focus on special analysis features. E.g. for a brand detection component specialized in detecting car brands a special analysis component could be developed and integrated. Also cloud-based 3rd party services could be used, e.g. a service like Shazam (shazam.com) which recognizes music titles. Of course, such integration services could prefer a lot from the availability of and compliance to standards within the media community and industry, something which the MediaMixer project is aiming at. One of those standards could e.g. be MPEG-21 which covers the lifecycle and interoperability of media resources within the industry.

7.1.3 Media Fragment Generation

As the result of the media resource analysis process the actual media fragments can be generated. These are either tracks (e.g. audio track, subtitle tracks in different languages), temporal fragments (e.g. shots, segments) or spatial fragments (e.g. bounding boxes); of course, spatial fragments always also have a temporal dimension. For a deeper discussion also see Chapter 3 Media Fragment Creation.

For later usage within re-use and mixing scenarios it is important to note that media fragments mainly serve two purposes: 1) the segmentation into meaningful parts, like shots and segments, and 2) the multimedia indexing of the

60 http://mpeg.chiariglione.org/standards/mpeg-21
video source. Both purposes result in the generation of media fragments but are in some respect contradictory: only the former can be the basis for later re-use in meaningful use case scenarios, i.e. result in “mixable” fragments. The purpose of the latter is to ever more precisely index the video resource that results in media fragments that can be shorter than 1 second. While these are not usable as media fragments in itself, they of course are important for finding relevant media fragments.

The different analysis results also have to be merged within a common representation format and consolidated and synchronized with the respect to the temporal and spatial dimensions. The result of the media fragment generation process is an index, which could be a file or stored within a specialized database.

With the generation of this index which contains the created media fragment information the first phase, the media resource analysis, is finished. Note that these media fragments are not yet media fragment URIs, as specified in the MF URI 1.0 standard.

7.2 Metadata Aggregation

The media fragment aggregation process covers the workflow chain from the initial generation of MF URIs, the generation of annotations and an enrichment via aggregation of further metadata by linking the media fragment annotations to Linked (Open) Data sources (see Figure 25). It also can include a manual editing process.

![Figure 25: Overview of the metadata aggregation process](image)

These stages are covered in more detail in the following subsections.

7.2.1 Metadata Conversion

The goal of the metadata conversion process is to generate Media Fragment URIs which are compliant to the above mentioned MF URI 1.0 standard. I.e. for each fragment as created by the analysis workflow, a single URI is created like `http://example.com/video.mp4#t=10,20`. These URIs do not carry any semantics in it, they just return a byte stream with the specified dimensions as part of the base video. In order to generate the domain of the URI (example.com in the case above), at this point it already should be known what the final location is. If this is not the case this can be overcome by employing abstract URIs or domain names, which can be resolved later by using a location resolution service, a mechanism similar to the usage of CRID in the broadcast context or to URI shortener services.

In order to be able to be enriched the media fragment URIs should be stored in a format which supports this best. This format is usually RDF based, as is the case within the LinkedTV project. Therefore, with respect to media fragment lifecycle management, an RDF repository is required as well as an API which supports retrieval and update of media fragments. The LinkedTV exposes such an interface this under `http://data.linkedtv.eu`.

For a description of the media fragment URI specification and the tools and technologies available see Chapter 2 Media Fragment Specification, Servers and Clients.

---

61 http://www.w3.org/TR/media-frags/
7.2.2 Annotation and Enrichment

The elements recognized by the analysis step are the basis for the annotation of the media fragments URIs. Each item which has been detected and thus gave rise to the creation of that particular media fragment is added as an annotation to the media fragment URI, usually in the form of RDF facts. The W3C standard for Open Annotation\(^{62}\) has been developed to standardize the format of these annotations.

The basic annotation step can also include the recognition of named entities which have been detected, by linking the entities to Linked Open Data (LOD) nodes, such as dbpedia. This can be done via the integration of Named Entity Recognition services. e.g. NERD as in the LinkedTV project (nerd.eurecom.fr) .

After the adding of basic annotations a further enrichment process tries to collect more facts which describe the media fragments in more detail. Based on the detected named entities, other LOD nodes can be linked with them, unstructured web pages can be added through internet search, or related media resources like images or video clips can be further added. Of course, also social media sources such as Twitter, Facebook or Slideshare can also be associated.

For a detailed description of the core technologies concerning annotation and enrichment cf. Chapter 4 Media Fragment Annotation.

7.2.3 Editing

In some contexts, before publishing the annotated and enriched media fragments, a manual editing step can be necessary. This is in particular the case within the TV broadcast scenarios, where the publisher wants to be in full control of what is being published and has to integrate an internal QA step. This editing is quite dependent on the later use case scenario. It basically includes removing of annotations, adding annotations manually or correcting annotations. Additionally searching and browsing functionalities are required as well as basic ontology editing facilities. With respect to related content DRM information for external media should be supported. Ideally, this does not only include information on license ownership, rights and billing, but also indicates country-specific rights information.\(^{63}\) In order to support this an editing tool is needed which acts as a client to the metadata repository and is able to read and write RDF-based annotations and to display the media fragments. All updates to the repository should contain provenance information, including motivation.\(^{64}\) Currently within the LinkedTV project such a tool is being developed, a first version will be ready by September 2013 (without support of DRM management).

Also, specific editing tools will be demonstrated in the context of the MediaMixer demonstration use cases. One example is the Editing tool of the Condat \textit{SmartMediaEngine}\(^{65}\) (Figure 26) which will be used in the Newsroom Use case.\(^{66}\)

\(^{62}\) http://www.w3.org/community/openannotation/
\(^{63}\) For the discussion of media fragments DRM management cf. Chapter 6.
\(^{64}\) For the relevant W3C provenance information see http://www.w3.org/2011/prov/wiki/Main_Page
\(^{65}\) http://www.condat.de/portfolio/medien/broadcast/portfolio/medien/broadcast/cms
\(^{66}\) http://community.mediamixer.eu/usecases/ucnewsroom/view
Another tool which is relevant in this context is Synote\(^67\), which has been described in Chapter 2.

After the final editing step the annotated and enriched media fragments are ready to be prepared for publishing and consumption.

### 7.3 Publishing and Consumption

The publishing and consumption of media fragments is the goal of the whole lifecycle steps before. However, this depends heavily on the respective use case scenarios and intended usage. For the different use cases the following main categories can be distinguished:

a) media fragment provider-based scenarios: these are scenarios where the license-holder of the media resources creates media fragments in order to license them to third-party stakeholders, which might be individual users, companies or organizations, including licensing and billing information.

b) media fragment exploitation-based scenarios: these are scenarios where publishers use media fragments (which might be their own or not) in order to provide specialized services, e.g. providing a media fragment portal which offers specialized media fragments on top of YouTube channels. Additionally, personalization and recommendation services may further enhance such services. The exploitation based scenarios do not necessarily require that media fragments itself are provided to the end user. E.g. a video remixing service could itself use media fragments which are combined to a while video clip, where the result of it is the whole clip rather than the parts of it. Within the TV-centered LinkedTV project the usage of media fragments mainly serves to annotate TV programs in such a way that annotations and related content can be displayed while broadcasting the program. Also the use case scenarios for VideoLectures.net and Newsroom Re-Use within the MediaMixer project fall into this category of exploitation-based usage.

\(^67\) [http://linkeddata.synote.org](http://linkeddata.synote.org)
c) media fragment end-user scenarios: these are scenarios where end-users themselves use media fragments, i.e. mainly search for them and then play them; in social media based applications this can also include enrichment by adding personal comments, ratings, annotations or recommendations.

As for the technology set all core technologies relevant here have been covered in the previous sections, as this requires Media Fragment URI compatible servers such as the Ninsuna Media Delivery Platform and compatible browsers or frameworks, as described in Chapter 2.

With respect to Media Fragment Lifecycle systems repositories and media asset management systems such as Fedora are of great importance. For a detailed discussion refer to Chapter 5. Beyond that, of course, all these components have to be integrated into greater production platforms which include an RDF Repository, DRM systems, streaming servers, content management systems and backend integration and workflow components such as an Enterprise Service Bus. These technologies, however, are not specific to media fragment lifecycle systems and as thus, beyond the core technology set.
8 Conclusions

Media fragment related technologies are still emerging, more often still used in research oriented contexts rather than commercial and production oriented ones. However, the basic standards are approved and technologies built on that can now mature and develop completely new applications and markets. The media fragment approach is a huge step in bringing technologies which are very well developed for the text-oriented parts of the Internet to the multimedia part thereof, enabling search and combining of multimedia content on a very fine grained basis.

As we have seen in the previous chapters, technologies and standards for media fragment based applications have been developed for all aspects of the whole media fragment lifecycle from analysis over creation to annotation and enrichment up to delivering, browsing and playing of media fragments. The importance of the media fragment technology lies not just in the extraction of small parts of videos, but rather in combining a whole set of approaches and technologies which are all on the cutting edge of multimedia and web technology, from video analysis to named entity recognition, linked data, semantic technologies and reasoning.

However, there is still much room for further research and innovation. To name just a few topics: video analysis is still more focusing on improving the results than on performance issues; real-time analysis of video streams would be a highly welcomed feature which to achieve still requires a lot of research and of course, availability of processing power. Automatic quality assurance is another topic: at this moment, media fragments and annotations are mostly just generated on base of the video analysis results, which can result in a lot of unusable fragments, which are difficult to clean up and correct automatically. In particular, to differentiate playable fragments from pure index fragments apart from shot detection is a difficult task. Sometimes it might be also appropriate to generate media fragments not only out of the video results, but by combining small fragments again to larger fragments containing meaningful segments; i.e. by generating synthesized fragments out of the “atomic” fragments which have been created at first hand. Also, the Media Fragment standard itself is very likely to be further developed, e.g. currently it only supports rectangular bounding boxes, in future versions it might also support circular and other spatial fragments. Further requirements are expected to come from the community.

But, as soon as the potential of the media fragment technology has been fully recognized by the creative industries and communities, and the technologies and processes are available, a lot of more usage scenarios and research and innovation topics will emerge. To foster this awareness is exactly the objective of the MediaMixer project.
9 References


[LinkedTV] LinkedTV EU Project, Deliverable D1.2 Visual, text and audio information analysis for hypervideo, first release 30.03.2012

http://www.linkedtv.eu/deliverables

[LinkedTVD5.1] LinkedTV EU Project, Deliverable D5.1 LinkedTV Platform Architecture, 2012; published at http://www.linkedtv.eu/deliverables

[LinkedTVD5.3] LinkedTV EU Project, Deliverable D5.3 First End-to-end platform, 2013; published at http://www.linkedtv.eu/deliverables


