Media Fragment Specification and Tools

Author: R. Troncy - EURECOM, France, email:Raphael.Troncy@eurecom.fr

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.

The mission of the W3C Media Fragments Working Group, which is part of W3C's Video in the Web activity, 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.

1 Purpose

We assume that media fragments are defined for "time-linear" media resources, which are characterised by a single timeline. Media fragments support addressing the media along two dimensions (in the basic version) and four dimensions (in the advanced version), see Figure 3.

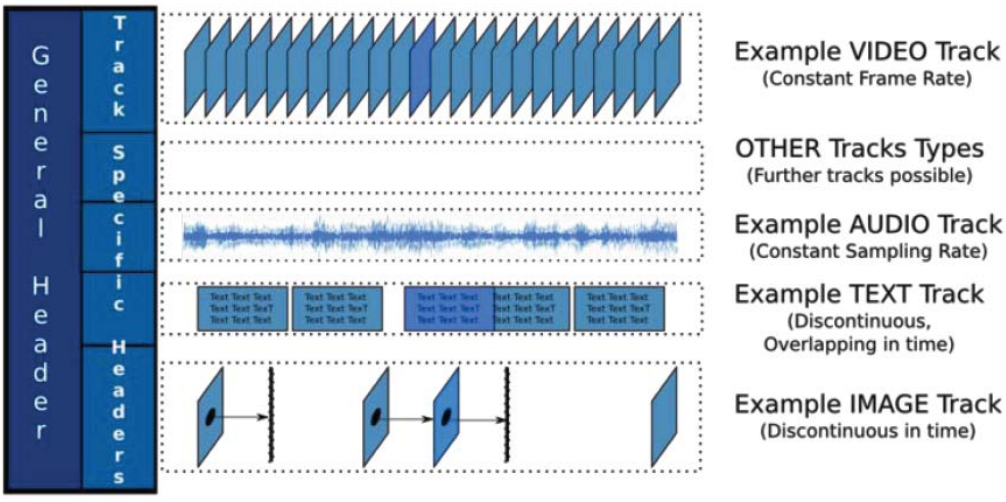

Figure 1: Media Fragments URI dimensions (or axis)

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 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.

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 four 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.

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.

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.

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

Use Cases

Tim does a keyword search on a video search service. That keyword is found in several videos in the search service's collection and it relates to clips inside the videos that appear at a time offset. Tim would like the search result to point him to just these media fragments so he can watch the relevant clips rather than having to watch the full videos and manually scroll for the relevant clips.

Silvia has a deaf friend, Elaine, who would like to watch the holiday videos that Silvia is publishing on her website. Silvia has created subtitle tracks for her videos (e.g. in WebVTT) and also a segmentation in chapters with unique identifiers on the clips that she describes. The clips were formed based on locations that Silvia has visited. In this way, Elaine is able to watch the videos by going through the clips and reading the subtitles for those clips that she is interested in. She watches the sections on Korea, Australia, and France, but jumps over the ones of Great Britain and Holland.

Yves is a busy person. He doesn't have time to attend all meetings that he is supposed to attend. He also uses his mobile device for accessing Web resources while traveling, to make the most of his time. Some of the recent meetings that Yves was supposed to attend have been recorded and published on the Web. A colleague points out to Yves in an email which sections of the meetings he should watch. While on his next trip, Yves goes back to this email and watches the highlighted sections by simply clicking on them. The various clips are played sequentially in a sort of jukebox.

Additional use cases are described in http://www.w3.org/TR/media-frags-reqs/

2 Method

In a well-known context where the MIME TYPE of the resource requested is known, various recipes are proposed depending on the dimension addressed in the media fragment URI, the container and codec formats used by the media resource, or some advanced processing features implemented by the User Agent. Hence, if the container format of the media resource is fully indexable (e.g. MP4, Ogg or WebM) and if the time dimension is requested in the media fragment URI, the User Agent will be in a position of issuing directly a normal RANGE request expressed in terms of byte ranges. On the other hand, if the container format of the media resource is a legacy format such as AVI, the Use Agent MAY privilege to issue a RANGE request expressed with a custom unit such as seconds and waiting for the server to provide the mapping in terms of byte ranges (Figure 4).

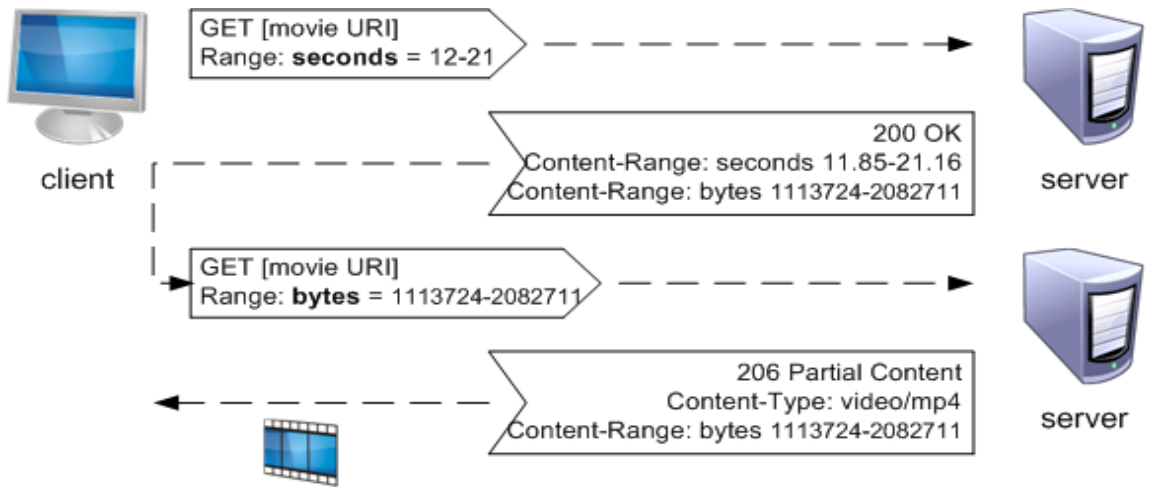

Figure 2: 4-ways handshake between a server and a user agent

The User Agent MAY also implement a so-called optimistic processing of URI fragments in particular cases where the MIME TYPE of the resource requested is not yet known. Hence, if a URL fragment occurs within a particular context such as the value of the @src attribute of a media element (audio, video or source) and if the time dimension is requested in the media fragment URI, the User Agent MAY issue directly a range request using custom units assuming that the resource requested is likely to be a media resource. If the MIME-type of this resource turns out to be a media type, the server SHOULD interpret the RANGE request. Otherwise it SHOULD just ignore the RANGE header (Figure 5).

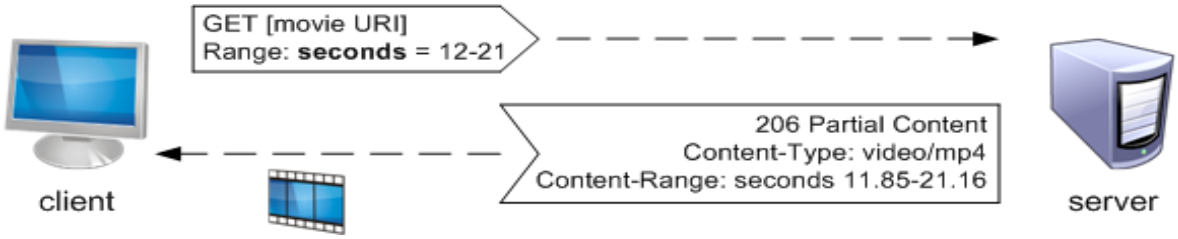

Figure 3: 2-ways handshake between a server and a user agent

3 Tools

Media Fragments URI Clients: a broad range of clients are capable of parsing and rendering media fragments URI. Those can be categorized into web browsers, custom HTML5 players or what is called polyfill libraries

- Web browsers supporting natively (at least partially) the Media Fragment URI specification:
  - Firefox (since version 9, version 28 as of April 2014)
  - Safari (since Jan 2012, announcement)
  - Chrome (since Jan 2012, announcement)
- Custom HTML5 players (non exhaustive list):
  - Ligne de Temps: http://ldt.iri.centrepompidou.fr/ldtplatform/ldt/
  - Media Fragments Enricher (MFE): http://mfe.synote.org/mfe/
- Javascript Libraries (or Polyfill):
  - mediafragment.js: https://github.com/tomayac/Media-Fragments-URI
  - xywh.js: https://github.com/tomayac/xywh.js

Media Fragments URI Servers: two different type of servers are being developed for demonstrating the capabilities of media fragments URI addressing and retrieval. One is based on Node.js while the other, Ninsuna, is a JAVA application.

Nisuna Media Delivery[1] 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). 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[2]. Its usage is as simple as run the command: `java -jar MFV.jar <mediafragment>`.

In addition, the major video sharing platforms have partial support for the Media Fragment URI specification, often just differing from a slightly different syntax. A thorough study of these platforms has been performed by Yunjia Li (Table 1).

[1] http://ninsuna.elis.ugent.be/ModelDrivenMediaDelivery
[2] http://ninsuna.elis.ugent.be/MFValidationService/resources/MFV.jar

Table 1: Media Fragment syntax supported in different video sharing platforms

Host	Example url	Fragment variable
56.com	http://www.56.com/u92/v_OTgwMTk4NDk.html#st=737	
Dailymotion	http://www.dailymotion.com/video/xjwusq&start=120 (is this a bug?) http://www.dailymotion.com/video/xjwusq?start=120	"start" query in seconds
Viddler	http://www.viddler.com/v/bb2a72e9?offset=12.083&secret=32758627	"offset" query in seconds
Vimeo	http://vimeo.com/812027#t=214 http://vimeo.com/812027?t=214	"t" query or hash in seconds
Tudou	http://www.tudou.com/listplay/H9hyQbAj4NM/2tzZHTtq4GA.html?lvt=30	"lvt" query in seconds
Youku	http://v.youku.com/v_show/id_XNjE2OTQ0MTI4.html?ev=5&firsttime=147	"firsttime" query in seconds
YouTube	http://www.youtube.com/watch?v=Wm15rvkifPc#t=120 http://www.youtube.com/watch?v=Wm15rvkifPc?t=120 http://www.youtube.com/watch?v=Wm15rvkifPc&t=1h9m20s http://www.youtube.com/watch?v=Wm15rvkifPc#t=1h9m20s	"t" query or hash in seconds or DDhDDmDDs format

Those video web sites are more or less popular, so let's consider the number of page views for each of those video sharing platforms according to http://www.websiteoutlook.com/. It shows how many videos, which users actually watch, can be further exposed by media fragments and furthermore, could be shared via social media, indexed by search engines and even linked to named entities at the **fragment level**.

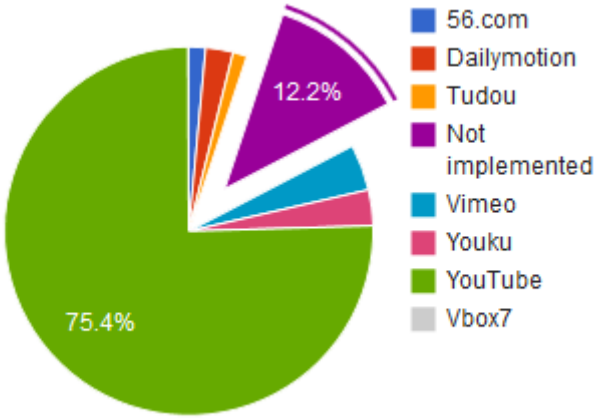

Figure 4: Percentage of video which are media fragments ready according to their audience

From the **Figure 6**, we can see that only 12.2% of video views are potentially not yet media fragments ready. This information can be interpreted in several ways:

- An end-user that wants to share only a part of a video with his friends has 90% chance of being able to do it.
- Most videos that we watched can be further indexed at the fragment level. This new SEO possibility will definitely bring more traffic to websites whoever implement it.

4 Usage and Deployment

We conducted a short analysis on Twitter in December 2013 to estimate how spread was the sharing action of media fragments URIs. Theoretically, any URL shared on Twitter could be a media fragment URI and ideally, one should monitor the twitter stream and check for each message whether a media fragment URI is included or not. However, this method is not realistic since it is difficult to automatically decide whether a URL in a message is a media fragment URI or not. Actually, some URLs may use a similar syntax while having nothing to do with media fragments (false positive cases), for example, http://www.example.org/1234#t=23. Consequently, we developed a program that monitors the twitter stream and looks for media fragments URI but restricted to links that point to one of 59 domain addresses that correspond to well-known web sites for sharing video content.

We run this crawling programme for 50 hours (from 12:00:00 GMT, 22nd Dec, 2013 to 14:00:00 GMT, 24 Dec, 2013) with the filter phrase "`youtube, dailymotion, vimeo, vbox7, viddler`". During those 50 hours, the crawler analyzed 5,779,858 Tweets, in which 5,269,742 Tweets include one or more URLs. A media fragment URI parser has been developed for detecting the media fragments encoded in those URLs. In total, there were 5,483,668 URLs processed out of which 32,796 URLs are valid media fragment URIs and 32,754 Tweets contain valid media fragment URIs. This means that only 0.6% of the video URLs shared from those websites via Twitter are media fragment URIs but this still represents an important volume.

5 Licensing and Contact

For more information and access to the tools described in this chapter, please refer to the table below.

Tool	**Link (URL)**	**Contact person**	**License**
Mediafragment.js	https://github.com/tomayac/Media-Fragments-URI	Thomas Steiner, tomac@google.com	CC0 1.0 Universal
xywh.js	https://github.com/tomayac/xywh.js	Thomas Steiner, tomac@google.com	CC0 1.0 Universal
Mediafragments-loose.js	https://github.com/yunjiali/Media-Fragments-URI-Loose	Yunjia Li, yl2@ecs.soton.ac.uk	CC0 1.0 Universal
Media Fragment Player	https://github.com/yunjiali/Media-Fragment-Player	Yunjia Li, yl2@ecs.soton.ac.uk	MIT

6 References

- Raphaël Troncy and Erik Mannens. Use cases and requirements for Media Fragments. W3C Working Draft, 17 December 2009, http://www.w3.org/TR/media-frags-reqs/.
- Raphaël Troncy, Erik Mannens, Silvia Pfeiffer and Davy van Deursen. Media Fragments URI 1.0 (basic). W3C Recommendation, 25 September 2012, http://www.w3.org/TR/media-frags/.
- Michael Hausenblas, Raphaël Troncy, Yves Raimond and Tobias Bürger. Interlinking Multimedia: How to Apply Linked Data Principles to Multimedia Fragments. In (WWW'09) 2nd Workshop on Linked Data on the Web (LDOW'09), CEUR Proceedings Vol. 538, Madrid, Spain, April 20, 2009
- Davy Van Deursen, Raphaël Troncy, Erik Mannens, Silvia Pfeiffer, Yves Lafon and Rik Van de Walle. Implementing the Media Fragments URI Specification. In 19th International World Wide Web Conference (WWW'10), Developer's Track, pages 1361-1364, Raleigh, North Calorina, USA, April 28-30, 2010
- Erik Mannens, Davy Van Deursen, Raphaël Troncy, Silvia Pfeiffer, Conrad Parker, Yves Lafon, Jack Jansen, Michael Hausenblas and Rik Van de Walle. A URI-Based Approach for Addressing Fragments of Media Resources on the Web. In Multimedia Tools and Applications, 59(2), pages 691-715, 2012
- Thomas Steiner and Raphaël Troncy. Tell me why! Ain't nothin' but a~mistake? Describing Media Item Differences with Media Fragments URI and Speech Synthesis. In (ICME'13) 1st International Workshop on Media Fragment Creation and Remixing (MMIX'13), San Jose, USA, July 15-19, 2013
- Yunjia Li. Are the Videos We are Watching Online Media Fragment Ready? October 17, 2013. http://afterglowlee.blogspot.fr/2013/10/are-videos-we-watched-online-media.html.
- Yunjia Li, Raphaël Troncy, Mike Wald and Gary Wills. Media Fragments Indexing using Social Media. In (ESWC'14) 2nd International Workshop on Linked Media (LiME'14), Heraklion, Greece, May 25, 2014.