# "Hyperlinking to time offsets: The temporal URI specification"

Dr Silvia Pfeiffer
Vquence, Xiph.org and Annodex.org

## Background

Since the year 2000 a project under the name of "**Continuous Media Web**", CMWeb, has explored how to make video (and incidentally audio) a first class citizen on the Web. One of the key challenges to overcome was the addressing of time offsets and time sections of a media file. This position paper explains the challenges and how the current specification has overcome them, such that the W3C Video Workshop can learn from the experiences and build on them for going forward.

## Requirements for temporal URIs (Uniform Resource Identifiers)

URIs are hyperlinks pointing at Web resources. With media files consisting of time-continuous data where the time component has a very strong meaning, URIs must be defined which take the time dimension into account. In fact, the spatial dimension is another component that should be addressable through URIs, but we are not addressing this in this paper.

A standard means of addressing **time offsets** into media Web resources is required so we can point people directly at a specific highlight inside a video.

In addition, a standard means of addressing **time segments** inside media Web resources is required such that we can extract and reuse such components to create dynamic mash-ups simply through providing a list of URIs, for example as a list of search results.

## The quest for a standard means of addressing

In the CMWeb project, we intended to develop a standard means of temporal URI addressing for any type of time-continuous Web resource. The only means of addressing subparts of Web resources are either through the query "?" or through the fragment "#" specification. Both of these can only be defined on a particular type of Web resource, e.g. the particular MIME type over which one has influence. This realisation implies that there is no means to prescribe for any and all video formats the way in which temporal URI addressing is being done.

For the CMWeb, we defined a temporal URI addressing format for the Ogg/Annodex family of media content. We are aware of different temporal URI addressing formats for MPEG-7, MPEG-21, and rtsp-based streaming mechanisms for other formats.

## Query or Fragment?

As we started to define the temporal URI addressing format, we naturally looked at the means in which offsets are being defined for HTML documents and tried to directly apply them to video. The HTML means consists of a fragment (“#”) identifier being added at the end of the resource URI, which will then encourage the Web client to load the full HTML page and undertake an offset action and to display the HTML document from the offset onwards.

Unfortunately, this is not simply transferable to video. RFC 3986 (URI RFC) specifies that URI fragments are interpreted client side, i.e. the Web infrastructure consisting of Web servers and Web proxies will generally ignore the "#" extension of URIs.

This means that in order to interpret a temporal offset, the user has to wait until all the data has been loaded until the offset point before the browser can start playing back the video. For videos that are hours long, this could potentially be a very long wait time and is a waste of bandwidth if one is only interested in 2 minutes.

Instead, by using the query “?” string, the offset action can be performed on the server and return to the Web client only the piece of video that the user is interested in, saving the user valuable waiting time and bandwidth.

Fragment addressing should still be utilised to jump around locally within a piece of already loaded video to avoid impacting the server.

For these reasons, in the CMWeb project, we decided to use both: a fragment and a query format for temporal URI addressing.

## Addressing of time offsets

There is no generically defined structure to URI queries. However, the format of a CGI query string where name-value pairs are separated by the special character "&" is a commonly used format. The Common Gateway Interface, or CGI, is a convention for external gateway programs to interface with information servers such as HTTP servers (see http://hoohoo.ncsa.uiuc.edu/cgi/). When defining a common manner in which temporal subparts of a Web resource can be addressed, it is important to be compatible with this common CGI format.

In the CMWeb project, we defined **temporal URI query strings** for Ogg/Annodex through a temporal query parameter “t=” which can take one or more intervals of time. Time is given with respect to specific time schemes. The default time scheme is "npt" (normal play time), in which a time point is specified in seconds through a floating point number with an arbitrary temporal resolution. Other time schemes specified at this point are the smpte time schemes and utc.

The BNF for a temporal URI query parameter is:

time-parameter = "t" "=" [ timescheme ":" ] time-intervalspec
time-intervalspec = time-interval ["," time-intervalspec]
time-interval = timespec ["/" timespec]

Examples of temporal URI queries are:
http://example.com/video.axv?t=npt:15.2 --- video.axv is transferred from 15.2 seconds into video.axv to the end of the file/stream.

**Temporal URI fragment specifiers** define a specific temporal view onto a Web resource consisting of one or more intervals of time. Again, time is given with respect to specific time schemes as defined below, "npt" being the default.

The BNF for temporal URI fragment identifier is (reusing the time-intervalspec from above):
temporal-fragment = time-parameter

Example:
http://example.com/video.axv#t=15.2 --- video .axv is transferred fully, but played back only from 15.2 sec ongoing.

## Addressing of time intervals

Occasionally, consumers have a need to virtually cut and paste pieces of video together to create a new piece. Specification of time intervals in URIs can provide this functionality.

Examples of time interval specifications are:

http://example.com/video.axv?t=15.2/18.7 ---
video .axv is transferred from 15.2 seconds into video.axv to 18.7 seconds; the default time scheme "npt" is used.
http://example.com/video.axv?t=15.2/18.7,23 ---
video.axv is transferred from 15.2s to 18.7s and from 23s to the end of the file/stream.
http://example.com/video.axv?t=15.2/18.7,17.4/30.1 --
video.axv is transferred from 15.2 seconds into video.axv to 30.1 seconds.

http://example.com/video.axa#t=15.2/18.7 --
a locally undertaken extract from a long video.

Examples for time-interval URI specifications with different time schemes are:

http://example.com/audio.anx?t=smpte-25:10:07:33:06/10:07:55:00
for a temporal interval between 36453.25s - 36475s.
http://example.com/audio.anx#t=npt:10:07:33.25
for a temporal interval between 36453.25s and the end of the file/stream.
http://example.com/audio.anx?t=clock:20021107T173045.25Z
for Thu Jul 11 05:30:45 UTC 2002 and a quarter seconds.

The draft specification of temporal URIs can be found here:
http://annodex.net/TR/draft-pfeiffer-temporal-fragments-03.txt

## Required server extensions

The delivery of videos from a time offset requires an extension to a HTTP server for delivering fully functional media files from an offset. For Ogg, there is an Apache module called mod_annodex, as well as Perl and Python bindings to provide this functionality.

The server needs to recompose a valid Ogg file from the Ogg headers and the data from the time offset without destroying the file integrity. There are fields to make sure this is possible in the Skeleton track for Ogg. More details can be found at http://svn.annodex.net/standards/draft-pfeiffer-oggskeleton-current.txt.

## A note on HTTP proxy servers

The temporal URI specification at http://annodex.net/TR/draft-pfeiffer-temporal-fragments-03.txt also describes a protocol, process and some novel HTTP request headers that are proposed to enable HTTP proxy servers to cache media files. To use proxies, one has to make use of the byte range specifications of HTTP.

Mapping of temporal URI queries to byte ranges can only happen on the origin server, being the only component that holds the base resource and can thus understand the relationship between time and bytes. The caching mechanism in HTTP/1.1 for byte ranges is based on the user agent requesting a URI with a "Range" request header that specifies the byte ranges. Thus, an additional agent-driven negotiation for delivery of the byte range mapping prior to the "Range" request is introduced to enable support of temporal URI queries.

The server can certainly cache the time-to-byterange mapping to avoid recalculations.

This specification has not been implemented or tests, but has been a theoretical experiment only to date.

## Summary

This document explained some of the challenges we solved when trying to define a standard means of addressing temporal offsets into media. The solutions all build on existing Web technology to make it possible to roll out over existing Web infrastructure.