INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC 1/SC 29/WG 11
CODING OF MOVING PICTURES AND AUDIO

ISO/IEC JTC 1/SC 29/WG 11
MPEG2017/m41072
July 2017, Torino, Italy

Source: **DMAG-UPC**
Status: **Proposal**
Title: **Description of DMAG-UPC's MPEG-G generation and decoding tool**
Authors: **Daniel Naro, Jaime Delgado, Silvia Llorente (Distributed Multimedia Applications Group – Universitat Politècnica de Catalunya)**

Table of Contents

1 Introduction

This document presents the status of the DMAG-UPC's tool to generate and read MPEG-G files. For the moment, it has been tested using the provided streams for input Id 05, for classes perfect match, match with N, match with SNP, match with indels and softclips.

It allows the following actions: combine a set of files in an MPEG-G container, list the blocks available in an MPEG-G container, read a region of the MPEG-G container, generate a new version of the MPEG-G container with certain regions encrypted.

2 Creation of a MPEG-G container

The tool uses a container definition written in XML that lists the contents used. We may highlight the definition of the `referenceSources` element that indicates to the library where to find the FASTA file containing the sources (the fields are not updated to reflect the status as of April 2017).

In order to create the container file, we need to extract different pieces of information from the streams, more specifically: where to find the MIT (indicated in the definition of the dataset), the numbers of access units for each reference, and for each stream the LIT.

As indicated in [1], we need to specify for each stream the contained information type. For the moment, we use the `id` attribute to this end.

The command to execute the creation of the file is:

```
encode <configuration_path> <output_path>
```

It should be noted that the path indicated as the source of the reference will be used during the decoding phase. As such, if the file path is absolute, it might fail on another machine, or if the path is relative but the file is not available, it will also result in a failure.

3 Listing the contents of a MPEG-G container

The command:

```
list_aus <path to MPEG-G file> <base file path>
```

lists the contents in terms of AUs: for each dataset group and each dataset in a file, a file is created in HTML format with the following name:

<base file path>_<datasetGroupIndex>_<datasetIndex>

The HTML document contains multiple tables, one per reference sequence, led by the id of the reference as a title. Each created table shows the values in the MIT, i.e. for each class (each row corresponds to a class) and for each Access Unit (each column corresponds to a new unit), we know the position of the first read in it.

Additionally, the encryption profile is also indicated after the hyphen: for example, in the screenshot of Figure 1, the first AU for reference 0 and class perfect match starts at 10005 is encrypted using profile 1. (Profile 0 indicates the absence of encryption).

0

10005-1 4346902-1 8326324-1 12352519-1 16606711-1 20134691-1 24078169-1 28197429-1 32291363-1 36125996-1 39978641-1 43704531-1 47498
12558-0 4463797-0 8513172-0 12410498-0 16920798-0 20269473-0 24175180-0 28209875-0 32320744-0 36145934-0 40008640-0 43761524-0 47577
10004-0 4346504-0 8326068-0 12352754-0 16606683-0 20134746-0 24078172-0 28197427-0 32290114-0 36126035-0 39978982-0 43704764-0 47498
10000-0 4346770-0 8326808-0 12354868-0 16607934-0 20137531-0 24079609-0 28197393-0 32290400-0 36126815-0 39979638-0 43705672-0 47497

1

11057-1 3812348-1 7411461-1 11201223-1 14705081-1 18258342-1 21815479-1 25420193-1 29188795-1 32805437-1 36307749-1 39871566-1 43314
77813-0 4289757-0 7437366-0 11322979-0 14798632-0 18394680-0 21940677-0 25621876-0 29236736-0 32877708-0 36375162-0 39877007-0 43371
11047-0 3812255-0 7411674-0 11202437-0 14705213-0 18258173-0 21815957-0 25420666-0 29189310-0 32805073-0 36307398-0 39870708-0 43314
10267-0 3815117-0 7412886-0 11201123-0 14706185-0 18259104-0 21816463-0 25420290-0 29188827-0 32808668-0 36308652-0 39874696-0 43315

2

60092-1 3347322-1 6791695-1 10372288-1 14191428-1 17793897-1 21124394-1 24492921-1 27871504-1 31203839-1 34734651-1 38155002-1 41673
88480-0 3528636-0 6886534-0 10408544-0 14277453-0 17804277-0 21657194-0 24514587-0 27932251-0 31263285-0 34996854-0 38272240-0 42016

Figure 1. Screenshot the HTML document detailing the position of the first read in each block (the title indicates the reference sequence, the row the class of data, and the column the Access Unit).

4 Reading content of a file

The content of a MPEG-G file can be returned in the form of a crude file indicating the position and the sequence (including the executed operations). For the moment, the output of this command is arbitrarily limited to the first two reference sequences to simplify the debugging.

This action is called using:

```
decode <MPEG-G path>
```

The output then takes the shape of:

Data in Datasetgroup_0/Dataset_0

Reference sequence 0

Access units 0

10000 TAACCCTAACCCTAACCCTGAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAAC
<->10027 CCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTCAACCCTAACCCTAACCACAACCCTAACGGCTAACCCT
10001 TAAACCCTAACCCTAAACCCTAACCCTAAACCCTAACCCTAACCCTAACCCTAACCCTAAGCCCTAAGCCCTAAGCCCTAAGCCCTAAGCCCTAAGCCCTAAGCCCTAACCCC
<->10357 TAACCCTAACCCTAACCCTAACCCTAAGACCTCATAGACCCTAACCCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAAGACCCTAACCCTAACCCT
10003 CCCTAACCCTAACCCTAACCCCTAACCCCTAACCCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAAGGCTAACCCTA
<->10265 CTATACCCTAACCCTAACCCTAACCCCAACCCCAACCCCAACCCCAACCCCAACCCCAACCCTCAACCCCTAACCCTAACCCTAACCCTAACCCTAACCCTA
10003 ACCCCTAACCCTAACCCTAACCCTAACCCTAACCCTAAGCCCTAAGCCCTAAGCCCTAAGCCCTAAGCCCTAAGCCCTAAGCACCTAAGCCCTAACCCCTAACCCTAACC
<->10310 ACCCCAACCCCAACCCTCAACCCCTCAACCCCTCAACCCTAACCCTACCCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCCTAACCCCTAACCCTAA
10004 CCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTA
<->10027 CCCTGAACCCTAACCCTCAACCCTAACCCTAACCCTAACCCTAACCCTAACCCCTAACCCCTAACCCTAACCCTAACCCTAACCCAACCCTAACCCTAACCCTA
10005 CTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAA

Green characters indicate insertions, red characters indicate deletions and orange ones softclips.

Despite not being depicted here, the presence of an N character would be marked in blue. The presence of an arrow prior to the position indicates the paired relationship with the previous read.

The output of this command can be filtered:

```
decode <MPEG-G path> [<class index={0,1,2,3 or 255 for all}> [<ref_id:start_pos-nd_pos>]]
```

5 Encrypting content of the file

With the following command, parts of the content of the file are encrypted:

```
encrypt <input path> <output path>
```

At the time of writing, the tool is set to encrypt all contents of class 0 streams, using a preselected password. Future versions will open access to more options.

6 API operations implemented

The WD part 3 defines a list of possible operations. We here list which operations are implemented, available and where.

Operation Name	**MPEG-G parsing library**	**MPEG-G decoding library**
GetHeader	Public interface	
GetHeaderField	Public interface	
GetMetadata	Public interface	
GetMetadataField	Public interface	
GetProtection	Public interface	
GetProtectionField	Public interface	
GetLabel	Public interface	
GetDatasetGroup	Public interface	
GetDataset	Public interface	
GetData	Public interface to MPEG-G decoding	Public interface
isSetField	Public interface	
ListMetadata	NA	
ListMetadataField	NA	
ListProtection	NA	
ListProtectionField	NA	
ListLabel	NA	
SearchMetadata	NA	
SearchMetadataField	NA	
SearchProtection	NA	
SearchProtectionField	NA	
SearchLabel	NA	
StreamData	NA	
AddHeaderField	Public interface	
AddMetadata	Public interface	
AddMetadataField	Public interface	
AddProtection	Public interface	
AddProtectionField	Public interface	
AddData	Public interface	
UpdateHeader	Public interface	

UpdateHeaderField	Public interface	
UpdateMetadata	Public interface	
UpdateMetadataField	Public interface	
UpdateProtection	Public interface	
UpdateProtectionField	Public interface	
UpdateData	Public interface (the complete data is updated, not a subset)	
Authorize	NA	
Verify	NA	
Conversion	Public interface (to an encrypted version in MPEG-G format)	
Beacon-like	NA	

Additionally to these operations, the MPEG-G decoding library gives additional operations:

Additional operations available in the public interface
LoadFileRegionAsMIT
MIT.InferAUFromPosition
LoadFileRegionAsLIT
LoadFileRegionAsStream
CreateStreamsCollection (for each of the classes)
StreamsCollection.CreateAccessUnit (each stream collection of a given class create access units of the same class)
CreateAccessUnitsMerger (merge sort on-the-fly reads returned by the different Aus)
GetNextRead
Read.GetPosition
Read.GetOperations (list of match, insert, deletion,.. operations)
Operations.GetSequenceAsString
Read.IsRead1
Read.HasPair
Read.GetPair
Read.GetPairPosition
Read.IsOnForwardStrand

7 Bibliography

[1] D. Naro, J. Delgado and S. Llorente, "M41070 Comments and issues in current MPEG-G Working Drafts (Parts 1 and 2)," Turin, Italy, 2017.