

UMID とその応用
～業界標準映像素材識別子 UMID の更なる活用に向けて～
(SMPTE UMID and its Applications)

SMPTE 30MR 技術委員会 UMID 応用調査グループ長(メタフロンティア合同会社) 柴田 賀昭

1. まえがき	1
2. UMID とは?	2
2.1. はじめに	2
2.2. UMID フォーマット	2
2.3. UMID の基本機能	2
3. UMID の誕生とその後の経緯	3
3.1. はじめに	3
3.2. UMID の誕生	3
3.3. VTR と UMID	3
3.4. ファイルベース機器と UMID	4
4. UMID の応用と課題	6
4.1. はじめに	6
4.2. UMID を用いた映像素材検索	6
4.3. UMID を介した複数 MAMS の連携	8
4.4. MEDIA SOA における UMID の応用	10
4.5. 解決すべき課題	12
5. SMPTE UMID 応用プロジェクト	13
5.1. はじめに	13
5.2. SMPTE UMID 応用プロジェクト	13
5.3. UMID 応用原理	14
5.4. UMID 解決プロトコル	18
5.5. MXF における UMID の応用	21
5.6. 拡張 UMID とその応用	24
6. まとめと替えて	28
REFERENCE	30

1. まえがき

この度、筆者らは、本映像情報メディア学会(ITE)から、2014 年度の映像情報メディア未来賞/フロンティア賞を拝受しました。これは、筆者らが 2011 年から取り組んできた表題に関わる活動の一環として、放送・映像メディア業界の業界標準映像素材識別子である UMID (Unique Material Identifier) [1]を活用する上で、全ての業務用映像

機器及び映像系アプリケーション(アプリ)が遵守すべき基本運用ルールを定めた「UMID 応用原理」(UMID Application Principles)を確立したこと、そしてこれを SMPTE (Society of Motion Picture & Television Engineers: 米国映画テレビ技術者協会)へ技術提案し、2014 年末に SMPTE RP 205 [2]なる標準規格の策定完了に至ったことを評価していただいたものです。

ここで SMPTE [3]ですが、これは 1916 年に発足した米国における映像技術関連の業界団体の一つです。しかし、これまで映像関連の技術開発の多くが米国で実施され、その相互運用性の確立に SMPTE が主導的役割を果たしてきたこと、そしてワールドワイドな映像産業における米国の主導的な立場なども相まって、特に映像制作技術の分野においては、事実上、SMPTE が国際標準規格の策定を担っています。

ところで、筆者らの当該活動の状況については、これまでも ITE 年次大会の場を利用して都度、報告して参りましたが[4-6]。しかしながら年次大会予稿といった限られた紙面では十分な説明ができなかったこと、更には、欧米に比べて我が国ではいわゆる「ファイルベース系」の研究開発活動がいま一つ活発でなかったことも相まって、これまで国内向けには、あまり積極的には情報発信に取り組んで参りませんでした。

今回、映像情報メディア未来賞/フロンティア賞の受賞に伴い、当該活動を紹介させていただく場を賜りました。そこで本稿では、この機会を最大限に活用して、単に今回の直接の受賞理由となった案件のみならず、現在進行中の他の UMID 応用関連案件も含めて、本稿執筆時点(2015 年 8 月)における筆者らの取り組みの全容を余すところなくお伝えさせていただきたいと思えます。

本稿は 6 章より構成されます。本章に続く第 2 章では UMID の基本事項を、また第 3 章では UMID の誕生とその後の経緯を紹介します。

次に第 4 章にて UMID 応用の具体例を紹介し、それらを実現する上で解決すべき三つの課題を提示します。

そして本稿の中心である第 5 章において、それらの課題を解決すべく筆者らが SMPTE 標準化コミュニティを舞台に取り組んできた「SMPTE UMID 応用プロジェクト」

を、その設立経緯も含めて概観した後、第4章で提示した課題に対する本稿執筆時点での検討状況を、課題毎に順を追って報告します。

最後に第6章にて、まとめをおこないます。

2. UMID とは？

2.1. はじめに

本章では、UMID の基本事項として、UMID フォーマット及びUMID に与えられた基本機能を紹介します。

2.2. UMID フォーマット

図1に、「基本UMID」と呼ばれるUMIDのフォーマットを示します[1]。

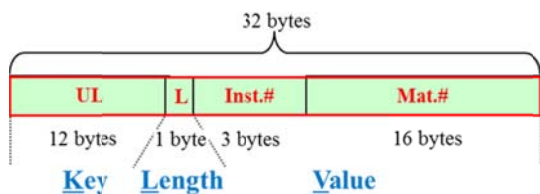


図1 基本UMIDのフォーマット

基本UMIDは、32byteのバイト列として構成され、先頭から順に、これがUMIDであることを示す12byteのユニバーサルラベル(UL)、1byteにてそれ以降に続くバイト列の長さを表すレングス(L)、このUMIDが付与された素材¹がオリジナルか否かを示す3byteのインスタンスナンバー(Inst.#)、そして最後に、このUMIDの大域的一意性(任意の時刻、場所において、それが唯一無二のものであること)を与えるマテリアルナンバー(Mat.#)となります。

ここでULは、SMPTEメタデータ辞書[7]に登録された固定値で、UMIDの場合、“06_h0A_h2B_h34_h01_h01_h01_h05_h01_h01_hxx_hyy_h”なる値をとります。ただし最後の2byteは、本UMIDが識別対象とした素材の種類や、Inst.#、Mat.#の生成方法に応じて、SMPTE ST 330 [1]が規定した値を設定することになります。

次にLですが、基本UMIDの場合、これに続くのは19byteであることから、“13_h”なる固定値をとります。なお、UL、L、そしてその後続くバイト列(Inst.#及びMat.#)はそれぞれ、Key、Length、Valueと呼ばれ、このKey-Length-Valueなる構造は、SMPTEが定義する全てのメタデータ関連技術の基盤となるものです[8]。

続いてInst.#ですが、2.3節で述べるように、その識別対象がオリジナル素材の場合はゼロ値(“00_h00_h00_h”)を、

¹ UMIDが識別対象とするのは、映像素材のみならず、音声素材、字幕素材、更にはそれらの複合素材なども含まれます。そこで本稿では以降、それらをまとめて単に素材と呼びます

そうでない場合は非ゼロの値を取ります。

最後にMat.#ですが、ここには、その生成時点において大域的一意性が保証された値を設定します。それは、例えばMACアドレスなど識別対象素材を生成した映像機器自体を大域的一意に特定する識別子と、当該素材を生成した時刻とを組み合わせることで16byteの1つの値とすることによって与えられます。このような大域的一意な値の生成方法は、例えばUUID (Unique Universal Identifier) [9]など、ICT分野では広く知られた技術です。

2.3. UMIDの基本機能

2.3.1. 大域的一意な素材識別子としてのUMID

UMIDの最も重要な機能は、その名の通り、それが付与された素材を大域的一意に特定するための素材識別子としての役割です。図2は、例えばカムコーダを用いた撮影にて新たな映像素材を生成した際、それに“U_i”なるUMIDを新規生成、付与した様子を示しています。

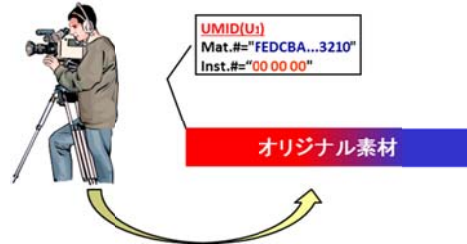


図2 素材識別子としてのUMID

図2において、UMID (U_i)のMat.#には、SMPTE ST 330 [1]に従い、当該カムコーダを一意に特定する識別子と当該素材の生成時刻から新たに生成された“FE_hDC_hBA_h...32_h10_h”なる値を、またそのInst.#には、その識別対象素材が今まさに生成されたオリジナル素材であることから、ゼロ値を設定しています。

この結果、当該素材はこの“U_i”でもって、大域的一意に特定されることとなります。

ところで、映像関連の識別子といえば、古くはISAN (International Standard Audiovisual Number) [10]から最近話題のEIDR (Entertainment Identifier Registry) [11]に至るまで、これまでも様々な技術が提案されてきました。

しかしその多くは、識別子(ID)の重複を避けるために専用のID登録サーバを設置し、それへの照会を通じて、大域的一意なIDを生成、付与しています。

これに対しUMIDは、2.2節で紹介したMat.#の生成方法が示すように、そのような専用ID登録サーバを設けなくとも、所定の生成アルゴリズム[1]に基づいて生成される限りにおいて、その値の大域的一意性が保証されています。そしてこのことは、必ずしもネットワーク環境

にあるとは限らない映像制作の様々な現場において、その場で生成された素材を大域的一意に特定するための ID を生成、付与する手段としては好適であるといえます。

2.3.2. 関連付けツールとしての UMID

もう一つの UMID の機能として与えられているのが、オリジナル素材への関連付けツールとしての役割です。図 3 に、図 2 で生成したオリジナル素材を部分的に切り出すことで派生素材を生成し、それに“U_i”なる UMID を生成、付与した様子を示します。

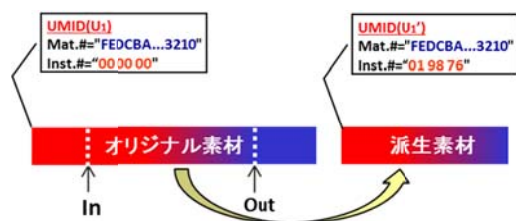


図 3 関連付けツールとしての UMID

図 3 において、UMID (U_i) の Mat.# は、オリジナル素材に付与された UMID (U₁) の Mat.# 値をそのまま継承して利用しています。他方、その Inst.# には、その識別対象がオリジナル素材でないことから、“01_h 98_h 76_h”なる非ゼロ値を設定しています。

すなわち、派生素材に付与される UMID にオリジナル素材の Mat.# 値をそのまま継承することで、それが当該オリジナル素材からの派生素材であることを示す訳です。

その結果、例えば派生素材に付与された UMID (U_i) に対して、その Inst.# 部分をゼロ値に置き換えることで、その生成元であるオリジナル素材を大域的一意に特定する UMID (U₁) が簡単に得られることとなります。

なお、ここで注意すべきは、そのようにして生成された UMID (U_i) は、その大域的一意性がもはや保証されないということです。なぜなら、そのような UMID は Mat.# を共有する以上、Inst.# の違いによって大域的一意性を確保する必要がある訳ですが、Inst.# に割り当てられたバイト数が高々 3byte であることから、Mat.# のように任意の時刻、場所において生成された Inst.# 値が決して重複しないことを理論的には保証できないからです。

換言すれば、UMID の基本機能として、2.3.1 節で紹介した大域的一意な素材識別子としての機能と、本節で紹介した関連付けツールとしての機能は排他的であって、両立できないということです。

ここで、非ゼロ値としての Inst.# 値の生成方法ですが、SMPTE ST 330 [1] には幾つかの Inst.# 生成方法が規定されていますが、その中で最も良く使われる方法は、先頭

の 1byte にオリジナル素材からの世代番号を、続く 2byte に例えば素材サーバなど所定の範囲内での Inst.# の重複を避けるための乱数を与えるというものです。

その具体例として、図 3 には、当該派生素材がオリジナル素材から直接生成されたことを示すべく Inst.# の先頭に“01_h”を設定し、続く 2byte には“98_h 76_h”なる乱数を設定した様子を示しています。

3. UMID の誕生とその後の経緯

3.1. はじめに

本章では、UMID の誕生と、その後の業務用映像機器における UMID の取り扱い状況について紹介します。

3.2. UMID の誕生

米国で地上デジタル放送が始まる直前の 1998 年 8 月、EBU (European Broadcasting Union: 欧州放送連合) と SMPTE の共同タスクフォース(TF)は、“EBU/SMPTE TF for Harmonized Standards for the Exchange of Programme Material as Bitstreams”なるタイトルの最終報告書を発表しました[12]。ここでの答申内容を簡単に紹介すると、放送・映像メディア業界においてデジタル化された素材のスムーズな交換を実現するために、

- 1) 業界特有の要求仕様に基づく素材ファイル書式を新たに開発し、それを業界標準規格にすること、
 - 2) ネットワーク空間に存在するファイルとしての素材を大域的一意に特定するための素材識別子スキームの統一化を図ること、
- でした。

この答申を受けて SMPTE が標準規格として策定したのが MXF (Material eXchange Format) [13] 及び UMID であり、それぞれ 2004 年及び 2000 年のことでした。

つまりこれこそが、今日「ファイルベース系」と呼ばれる映像制作技術の本格的な始まりであったといえます。

3.3. VTR と UMID

3.3.1. HDCAM™ における UMID 機能の実装

UMID が SMPTE 標準規格となった 2000 年は、まさに業務用 VTR の全盛期でした。そこで当時、筆者らが在籍していたソニー株式会社(ソニー)では、筆者らが中心となって VTR における UMID の取り扱い方法を精力的に検討し、その結果を踏まえ、同社の当時の主力製品であった HDCAM™ シリーズ[14] に対して、UMID の取り扱い機能を実装しました。

図 4 は、HDCAM™ シリーズで実現された VTR にお

ける UMID の取り扱いを模式的に示したものです。

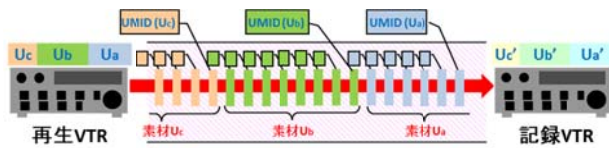


図 4 VTR における UMID の応用

そもそも VTR において素材をどう定義するかですが、ここでは一回の記録開始/終了(REC Start/Stop)にて得られる連続したフレーム列を単一の素材とみなし、これに一つの UMID を付与することとしました。この結果、複数の記録操作をおこなった場合、一本のテープ上には複数の素材が連続して存在することになります。

図 4 左部にある VTR の上部には、一本のテープ上に 3 つの素材を連続して記録し、それぞれ記録順に、“U_a”、“U_b”、“U_c”なる UMID を付与した様子を示しています。

3.3.2. UMID の記録と伝送

ところで、テープ上に記録された素材に対して VTR がある瞬間においてアクセス可能なのは、まさにそれが読み出し対象としたフレームが記録された部分のみです。他方、素材に付与された UMID は、当該素材の記録形式に関わらずいつでも簡単に読み出すことができなければ、素材識別子としての役割を果たし得ません。

この問題を解決するため、テープ上に記録された素材においては、それを特定する UMID を、当該素材を構成する全てのフレームに付与してテープに記録することとしました。これより、VTR が当該素材のいずれのフレームをアクセスしても、それを含む素材に付与された UMID を知り得ることが可能となります。

このように記録された UMID は、素材を VTR 再生、SDI (Serial Digital Interface) 出力した際には、図 4 中央部に示したように、SDI の VANC (Vertical ANCillary data) [15]にそのままのかたちで挿入され、出力されます。

一方、これを受けた記録 VTR は、通常は 2.3.2 節で紹介した機能に従い、入力フレームに付与された UMID (例えば“U_a”)からその Mat.#を抽出し、新たに生成した非ゼロの Inst.#を加えて構成した新たな UMID (同、“U_a”)を、当該フレームと共にテープに記録します。

従って記録 VTR のテープ上には、図 4 右部にある VTR の上部に示したように、“U_a’”、“U_b’”、“U_c’”なる UMID が付与された 3 つの素材が連続して記録されることとなります²。

² もうひとつの選択肢としては、入力素材に関わらず記録 VTR 側で新たな素材を生成するとの考えの元、所定の記録開始(In 点)、終了(Out 点)を設定して、2.3.1 節で

一般にテープ上の素材は非可逆圧縮(lossy compression)されたかたちで記録され、ダビングやインサート編集など SDI を介した画像処理の際には、これが一旦ベースバンド信号に伸張された後、再び非可逆圧縮されてテープに記録されます。この結果、映像制作に伴う画像データの圧縮伸張の繰り返しが画質の著しい劣化につながることから、そのオリジナルな段階から施された画像処理の回数(いわゆるダビング時の「世代番号」に相当)の適切な管理が重要となります。

本節で紹介した UMID 機能を用いると、2.3.2 節で紹介したように、その Inst.#の先頭バイトに当該世代番号をフレーム単位で記録することができます。従って、例えば映像制作の際に用いる素材の世代番号に上限を設けることによって、最終成果物の品質を一定レベルに保つよう素材を管理することが可能となります。

3.3.3. 拡張 UMID

3.3.2 節で議論したフレーム毎に付与される UMID は、2.2 節で紹介した基本 UMID 形式の替わりに、拡張 UMID と呼ばれる形式をとることも可能です。この場合、基本 UMID に加えて、それが付与されたフレーム(のオリジナルデータ)を「いつ」、「どこで」、「だれが」最初に生成したかといった情報を追記することも可能となり、UMID 応用の有用性が更に広がります。

拡張 UMID 及びその応用については、5.6 節にて改めて詳しく紹介します。

3.4. ファイルベース機器と UMID

3.4.1. XDCAM™ における UMID 機能の実装

3.3 節で紹介したように、VTR 全盛の時代からその応用検討が進められ、一部の取り扱い機能が実装された UMID でしたが、その「本来あるべき姿」である大域的一意な素材識別子としての UMID 運用が本格的に実現できたのは、素材をファイルとして取り扱ういわゆる「ファイルベース」機器からであり、筆者にとってその端緒となったのが、その後のソニーの主力製品となった XDCAM™ シリーズ[16]の映像記録フォーマットである Professional Disc™ 仕様の開発に取り組んだことでした。

XDCAM™ とは、撮影した映像素材を光ディスク(その後は SSD(Solid State Disc)へも展開)に MXF ファイルの形式で記録する同社の業務用映像機器の総称です。2003 年に最初の SD (Standard Definition)対応モデルが発売され、

紹介した素材識別子としての UMID を新規生成、付与するという方法もあります。

その後の HD (High Definition)対応モデルを経て、本稿執筆時点では 4K での撮影、記録に対応したモデルも販売されています。

ところで UMID は、その誕生の経緯から、MXF ファイルとしての素材を大域的一意に特定するための必須項目として MXF 規格に採用されました[13]。その詳細については改めて 5.5 節で紹介しますが、簡単に言えば、ある素材が MXF ファイルとして与えられた場合、それを大域的一意に特定するための UMID³が、当該 MXF ファイルのヘッダ部分に埋め込まれているということです。

3.4.2. XDCAM メタデータの概要

さて、VTR と異なりランダムアクセスが可能であることを最大の特徴としたファイルベース機器では、そのアクセスのヒントを与えるためのメタデータの役割が益々重要となりました。そこで Professional DiscTMでは、MXF ファイルとして記録した素材に加えて、素材の撮影日時や再生時間長、更にはタイトルや撮影内容の概要などを XML (eXtensible Markup Language) [17]で記述した NRT (Non-Real Time Metadata)と呼ばれるメタデータと、タイムコードなどフレーム毎に変化する情報を列挙して BiM 方式[18]で記述した RT (Real Time Metadata) と呼ばれるメタデータとを併せて生成、記録することとしました。

そして NRT、RT と、それらが記述対象とした素材との関連付けを、当該素材としての MXF ファイルに付与された UMID への参照にて実現しました。

図 5 に、Professional DiscTM仕様が規定した XDCAM メタデータの概要図[19]を再掲します。

図 5 において、二つの「AV clip」はそれぞれが、MXF ファイルとして生成、記録された一つの素材を示します。そして各々の素材には、それを大域的一意に特定するための UMID が生成され、対応する MXF ファイルのヘッダ部分に埋め込まれています。

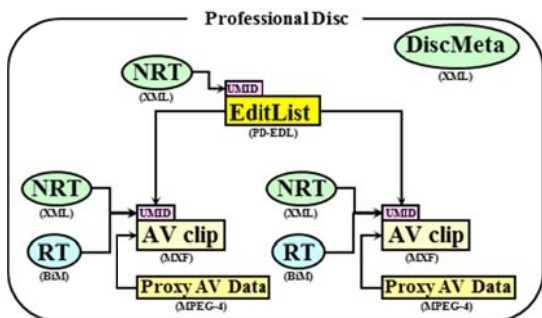


図 5 XDCAM メタデータの概要[19]

また、各々の素材を記述対象としたメタデータである NRT 及び RT が当該素材としての MXF ファイルと同時に生成、記録され、それらが同 MXF ファイルのヘッダ部分に埋め込まれた UMID を参照することで素材とメタデータとの関連付けをおこなっており、図 5 ではこれを、NRT 及び RT からその記述対象である AV clip に付与された UMID へ向けた矢印にて表しています。

なお、XML で記述されたメタデータと MXF ファイルとしての素材との関連付け方法については、4.2.1 節にて改めて詳しく紹介します。

ところで、Professional DiscTM仕様では、本素材と呼ばれる MXF ファイルとして生成、記録された素材と並行して、プレビューあるいはオフライン編集用途を目的に、当該本素材と内容は同一ながら解像度(従ってデータ量)を大幅に落としたいわゆるプロキシ素材も同時に生成、記録することを可能としており、図 5 ではこれを「Proxy AV Data」にて示しています。ただし、このプロキシ素材もまた MXF ファイルの形式で記録されることから、本素材とプロキシ素材との間の関連付けについては、後者を前者の派生素材とみなすことで、2.3.2 節で紹介した関連付けツールとしての UMID⁴の機能を採用しました。

更に Professional DiscTM仕様では、記録済み本素材の所望の部分だけを切り出して連続再生するいわゆる「クリップリスト」編集再生という機能があり、これを実現する手段として、EditList と呼ばれる編集記述データを導入しました。ここでは編集再生対象とした本素材をその UMID にて、またその中の所望の部分(タイムコードを用いた)In/Out 点にて指定した「サブクリップ」と呼ばれる記述データが複数、所望の再生順に列挙されており、これらを順次読み込んでシームレスに再生することで、クリップリスト再生機能を実現しています。

そして Professional DiscTM仕様では、この EditList もまた(クリップリスト編集機能によって得られる)新たな素材とみなし、それを大域的一意に特定するための UMID を新たに生成、付与した他、それを記述対象とした NRT も同時に生成し、当該 UMID を介した関連付けをおこなっており、図 5 ではこれを、NRT から EditList に付与された UMID へ向けた矢印にて表しています。

なお、図 5 右上部にある「DiscMeta」は、素材やメタデータを記録した光ディスクなど記録媒体そのものに関する情報を記述した XML 形式のメタデータであって、UMID とは特に関係はありません。

³ 2.3.1 節に従い、新規生成した Mat.#と、ゼロ値の Inst.#との組み合わせとして構成されます。

⁴ 2.3.2 節に従い、本素材に付与された UMID から継承した Mat.#と、新規生成した非ゼロ値の Inst.#との組み合わせとして構成されます。

3.4.3. 大域的一意な UMID 運用の実現

3.4.1 節では、その「本来あるべき姿」である大域的一意な素材識別子としての UMID 運用を、XDCAM™ シリーズで本格的に実用化したことを述べました。これは換言すれば、XDCAM™ シリーズが取り扱う全ての素材の各々が、それに付与された UMID によって大域的一意に特定できることを保証したということに他なりません。

この UMID の大域的一意性の保証ですが、一見それ程難しくもないようにも見えますが、実はそれは見掛けほど簡単な話ではありません。

確かに SMPTE ST 330 [1]に従えば、大域的一意な UMID は容易に生成できます。しかしこれは、当該規格に従って自らが今まさに生成した UMID が大域的一意であることを保証しているに過ぎず、見ず知らずの第三者が生成した MXF ファイルに付与された UMID の大域的一意性を必ずしも保証するものではありません⁵。

特に、既存の MXF ファイルをコピーしその内容を上書き変更することで新たな素材を得た場合、得られた MXF ファイルはコピー元のそれとは別物となりますから、異なる UMID にて両者を明確に区別する必要があります。しかしこれは、上書き変更で得た MXF ファイルに対して、そのヘッダ部分に含まれていた UMID を、別途新たに生成した UMID によって意図的に置き換えることが必要と言っていることに他なりません。

すなわち、それを維持するためのメンテナンス対応に能動的に取り組まなければ、UMID の大域的一意性は簡単に破られてしまうということです。

XDCAM™ シリーズでは、第三者が生成した MXF ファイルを例えば FTP (File Transfer Protocol) を用いて取り込む場合、原則として元々それに付与されていた UMID はその大域的一意性が信頼できないものとみなし、その、自ら新規生成した信頼できる UMID への置き替えを実施しています。

また記録済み MXF ファイルの上書き編集(いわゆる「破壊編集」)を実施した場合、元の MXF ファイルのコピーがどこかに存在することを前提に、上書き編集後の MXF ファイルに元々付与されていた UMID を新たに生成した UMID に置き換えた上で、それ以前に当該 MXF ファイルを参照していた NRT や RT との関連付けを再構築しています。

更にこの UMID の大域的一意性維持のためのメンテナンス対応は、その機能的役割から、より頻繁にその内容

が変更されるであろう EditList についても例外ではなく、その結果、例えば既存の EditList の内容をほんの僅か変更しただけであっても、その保存時において新たな UMID を生成、置き換えた上で、それ以前に当該 EditList を参照していた NRT との関連付けを再構築しています。

このような UMID のメンテナンス対応を忠実に実装した結果、少なくとも XDCAM™ シリーズの製品群で閉じられたファイルベース制作・送出システムの中では、そこに存在する任意の素材に付与された UMID は、常にその大域的一意性が完全に保証されることとなります。

そして、筆者らが Professional Disc™ 仕様開発に取り組んだこの経験こそが、業界標準映像素材識別子である UMID を「本来あるべき姿」である大域的一意な素材識別子として運用するための基本運用ルールすなわち「UMID 応用原理」の確立と、それを全ての業務用映像機器及び映像系アプリに遵守させるための SMPTE 標準規格の実現に向けた礎となった次第です。

4. UMID の応用と課題

4.1. はじめに

本章では、特にその「本来あるべき姿」である大域的一意な素材識別子としての UMID の様々な応用事例を紹介し、それらを実現するための課題について述べます。

4.2. UMID を用いた映像素材検索

4.2.1. 素材とメタデータの関連付け

3.4.2 節で紹介したように、XDCAM™ シリーズでは、MXF ファイルとして生成された素材と並行して XML で記述されたメタデータを生成し、当該素材を大域的一意に識別すべく MXF ファイルのヘッダ部分に埋め込まれた UMID を介して、両者の関連付けをおこなっています。

図 6 は、この様子を改めて模式的に示したものです。なお、同図で示したメタデータは、Professional Disc™ 仕様が規定したのではなく、あくまで本章での事例紹介のために準備したものであることにご注意下さい。

図 6 では、それ自身を大域的一意に特定する UMID として“U_A”が付与された(映像)素材と、当該素材の内容を記述したメタデータが、それぞれ MXF 及び XML ファイルとして生成、記録されたこと、そして当該 XML ファイルにおいて、その TargetMaterial 要素がもつ umidRef 属性が UMID 値“U_A”を指定することで、このメタデータが当該素材を外部参照している様子が示されています。

そして当該メタデータの記述内容ですが、当該素材に対して「大リーグベースボール」なるタイトルを付与し

⁵ 事実、そのファイルヘッダ内の UMID 領域がゼロ値で埋められたような MXF ファイルが、映像制作の現場に流通しています。

たこと、またそれに含まれるハイライトシーンを、各々のシーンの先頭から抽出したキーフレームの注釈として記述したものを列挙し、その一例として、「イチロー」が「ホームラン」を打ったシーンを、“00:12:34:15”なるタイムコード(hh:mm:ss:ff 形式)が付与されたキーフレームの注釈として記述しています。



図 6 素材とメタデータの関連付け

このように、メタデータに対して、それが記述対象とした素材を一意に特定するUMIDへの外部参照記述を持たせることで、両者の関連付けを論理的に表現することが可能となります。

4.2.2. UMIDに基づく素材検索

図 6 で紹介したようなメタデータを導入する最大の目的は、高効率な素材検索の実現です。図 7 は、放送局内におけるファイルベース制作・送出システムにおいて、その典型的な一例を紹介したものです。

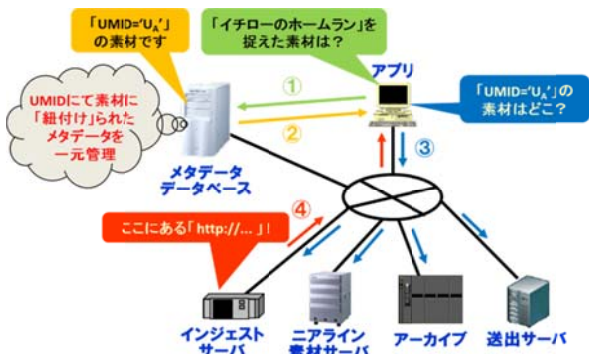


図 7 UMIDに基づく素材検索

図 7 で示したように、一般的にファイルベース制作・送出システムは、回線収録など外部から素材を取り込むための「インジェストサーバ」、映像制作で近々に必要な素材を一時的に保管するための「ニアライン素材サーバ」、完成した映像プログラム(完パケ)をオンエア送出するための「送出サーバ」、そして送出済み完パケを長期保存するための「アーカイブ」といった様々な「素材サーバ」

から構成されています。

そして、そのデータサイズが数百 GB を超えることも珍しくない素材ファイルに対してメタデータのデータサイズは高々数百 KB 程度に留まるなど、両者のデータサイズが大きく異なることから、通常、それらは分離して管理されることとなります。

図 7 は、ネットワークに接続された各種素材サーバが様々な素材を格納し、他方でそれらの内容を記述したメタデータを素材から分離して専用のメタデータ・データベースに集め、まとめて一元管理している様子を示しています。そしてこの時、ある素材とその内容を記述したメタデータとを論理的に「紐付け」ているのが、図 6 で紹介したメタデータから素材へのUMID参照です。

いま、このようなファイルベース制作・送出システムにおいて、例えば映像編集ツールなどある映像系「アプリ」が「イチローのホームラン」シーンを捉えた素材を欲したとします。この時当該アプリは、例えば「イチロー」、「ホームラン」などを検索キーとして、その旨メタデータ・データベースに問い合わせします(図 7 中①)。

メタデータ・データベースには、例えば図 6 中のXMLファイルで示したようなメタデータがまとめて一元管理されていますので、入力された検索キーを手掛かりに例えばテキスト検索を実施することで当該XMLファイルに相当するメタデータを取り出します。そして当該メタデータの記述内容からそれが記述対象とした素材すなわち当該メタデータの外部参照先を検出することで、それが“U_A”なるUMIDで大域的に一意に特定される素材であることを見出し、その結果を問い合わせ元のアプリに戻します(図 7 中②)。

これより、当該アプリは所望の素材が“U_A”なるUMIDで特定される素材であることを知り得ます。

しかし、残念ながらUMIDそれ自体には、所望の素材の格納実体である素材ファイルがどこにあるかといった情報までは含まれていません。そこで当該アプリは、このシステムを構成する全ての素材サーバに対して、“U_A”で特定される素材を格納しているかどうかを問い合わせることとなります(図 7 中③)。

その結果、図 7 の事例では、「インジェストサーバ」が当該素材を格納していたことを表明し、その旨を問い合わせ元のアプリに回答します(図 7 中④)。この際、当該インジェストサーバが、ファイルとしての当該素材にアクセスする手段であるURL (Uniform Resource Locator) も併せて当該アプリに通知することで、当該アプリは、例えばFTPを用いて当該素材ファイルを手にダウンロード

ードできる、すなわち当初の目的であった所望の素材の入手が適うこととなります。

4.3. UMID を介した複数 MAMS の連携

4.3.1. MAMS とは？

ファイルベース機器を用いた映像制作が広がるにつれ、多くの業界ベンダが素材管理のための MAMS の導入を積極的に提案しています。

ここで MAMS (Media Asset Management System) とは、一般的には、映像、音声などのコンテンツをそれらのメタデータと共に蓄積、管理し、再利用を始めとしたそれらの有効活用を支援するシステムの総称です。

ただここでは、MAMS とは、映像制作の現場における素材の再利用を支援すべく、ファイルとしての素材と、所望の素材を効率よく検索するためのメタデータとをまとめて一元管理したツールであるとお考え下さい⁶。

4.3.2. 複数 MAMS の連携

さて、効率的な素材管理のために管理対象とする素材に一意な素材識別子(素材 ID)を付与し、これを主キー(Primary Key)として運用することは、多くの MAMS にとっての常套手段です。ただ当然のことながら、素材 ID の一意性は、その MAMS が管理する範囲内で保証できれば十分な訳ですから、ここでは必ずしも大域的一意な素材 ID を必要とするものではありません。

しかし、これはあくまで一つの MAMS の中で閉じた場合の話でして、もし互いに独立した複数の MAMS を連携運用したい場合、それらが取り扱う素材 ID の整合性が問題になってきます。その一例を図 8 に示します。

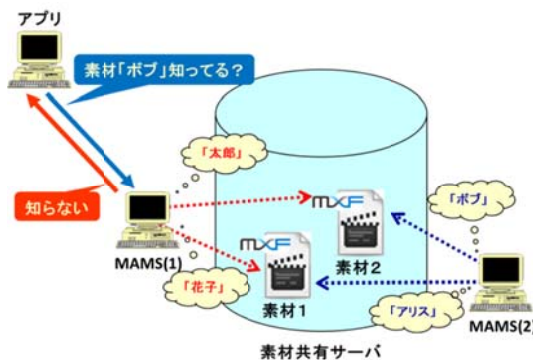


図 8 複数 MAMS の連携 - 現状

図 8 は、「MAMS(1)」、「MAMS(2)」なる二つの MAMS が素材サーバを共有し、それに格納された「素材 1」、「素材 2」なる二つの素材をそれぞれが独立して管理してい

る様子を示したものです。ここで MAMS(1)、MAMS(2) は、それぞれが独自に保有する素材 ID スキームに従い、例えば「素材 2」に対して、MAMS(1)は(日本人名風の「太郎」なる素材 ID を、また MAMS(2)は(米国人名風の「ボブ」なる素材 ID を生成し、付与しています。

いま、ある素材を所望する映像系「アプリ」がメタデータ検索の結果、当該素材を識別する素材 ID が「ボブ」であることを知り得て、偶々近くに存在した MAMS(1) に対して、4.2.2 節の図 7 中③で示したような問い合わせを実施したとします。

これに対する MAMS(1)の反応ですが、MAMS(1)にとって「ボブ」なる素材 ID は未知のものですから、当該アプリが求めているのが実は自らも管理している「素材 2」であったにも関わらず、MAMS(1)は当該アプリに対して「知らない」と回答せざるを得ません。

さらに、各 MAMS はそれぞれ独自の素材 ID スキームを運用している訳ですから、例えば異なる素材に対して、それぞれが偶々「ケン」⁷なる同じ素材 ID を生成、付与してしまうことも想定されます。すなわち、何か特別なことをしない限り、両 MAMS を連携運用する範囲において、素材 ID の一意性を保証する術はありません。

このような問題の解決こそ、まさに業界標準映像素材識別子としての UMID が果たすべき役割に他なりません。

すなわち UMID は、各々の MAMS が保有する素材 ID スキームとは無関係に生成することが可能であり、またその生成時における大域的一意性が保証されていますので、複数の MAMS が連携運用する範囲においても、それぞれが UMID として生成、付与した素材 ID は、決して重複することはありません。

さらに図 8 で提起された問題に対しても、それぞれの MAMS が図 9 に示したような対応をおこなうことで、その問題を解決することが可能となります。

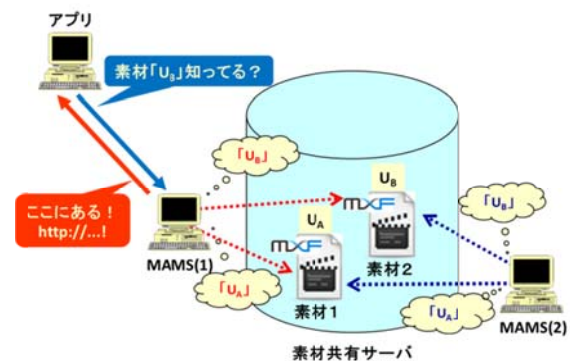


図 9 複数 MAMS の連携 - あるべき姿

⁶業界ベンダによっては、これを PAM (Production Asset Management) システムと呼ぶところもあります。

⁷ 日米に共通して広く用いられる人名(通称)のひとつです。

図 9 は、「MAMS(1)」と「MAMS(2)」が共有する素材サーバに「素材 1」と「素材 2」が格納され、それぞれ“U_A”、“U_B”なる UMID が素材 ID として予め付与されている様子を示したものです。ここで当該素材 ID は、それが SMPTE ST 330 [1]に従って生成された UMID である限り、両 MAMS を含め、いずれの業務用映像機器あるいは映像系アプリが生成、付与したもので構いません。

そして図 9 ではさらに、MAMS(1)及び MAMS(2)がそれぞれの素材管理において、各素材に付与されていた UMID を抽出し、当該 UMID を、各々が管理するメタデータの一項目として当該素材と共に管理している様子も示しています。

このような、事前に素材に付与された UMID を抽出、管理する機能が適切にサポートされたならば、いま、例えば「素材 2」を所望する映像系「アプリ」がメタデータ検索の結果、当該素材を識別する UMID が“U_B”であることを知り得た場合に、当該アプリが MAMS(1)、MAMS(2)のいずれに対して“U_B”で一意に特定される素材の有無を問い合わせたとしても、所望の結果を得ることが可能となります。

換言すればこれは、MAMS(1)と MAMS(2)の連携運用において、各素材に付与された UMID が、まさに両 MAMS 間の共通素材 ID としての役割を果たしているということに他なりません。

ここで注意すべきは、本節で紹介した UMID 管理機能のサポートは、決して、それぞれが独自に保有する素材 ID スキームを破棄し UMID のそれに入れ替えることを求めている訳ではないということです。つまり、ここで各々の MAMS に求められているのは、基本的には、独自に保有する素材 ID スキームはそのままに、UMID を、対応する素材と共に管理すべきメタデータ項目の一つとして追加するといったことに過ぎません⁸。

そして、両 MAMS がこのような UMID 管理機能を適切にサポートすれば、MAMS の標準機能としての(それが提供する独自の素材 ID を介した)ある素材に関する複数のメタデータ項目間の相互参照を通じて、図 9 で示したような MAMS(1)の応答、すなわち問い合わせを受けた UMID が一意に特定する素材をアクセスするための URL を通知する、といったことが可能になる訳です。

4.3.3. システム内共通素材 ID としての UMID

4.3.2 節では、「MAMS(1)」、「MAMS(2)」なる二つの

MAMS を連携運用する場合において、両者にまたがって運用可能な共通素材 ID としての UMID を取り扱うための UMID 管理機能を紹介しました。

ところで、ファイルベース制作・送出システムを構成する業務用ファイルベース映像機器や映像系アプリ(ファイルベース機器/アプリ)は、多かれ少なかれ何らかの形で、それらが取り扱い対象とする素材の管理機能をサポートしていることが一般的です。

そこで 4.3.2 節の議論を拡張し、当該システムに含まれる全てのファイルベース機器/アプリが同節で紹介したような UMID 管理機能をサポートしたならば、これはまさに、UMID が当該システム内の共通素材 ID としての役割を果たすということと等価となります。図 10 は、これを模式的に示したものです。

図 10 では、当該システムを構成する一連の基本的なファイルベース機器/アプリがネットワークに接続され、その間の素材のやり取りを、システム内での共通素材 ID としての UMID が付与された MXF ファイルの交換にて実施している様子を示しています。

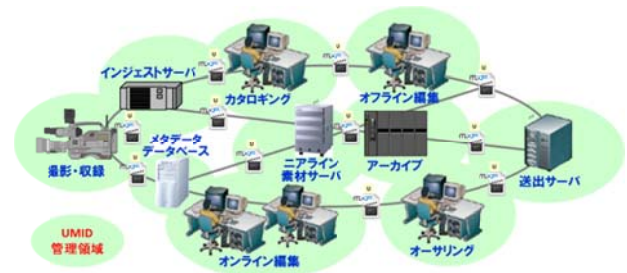


図 10 システム内共通素材 ID としての UMID

そして、当該システムの各々の構成要素が「UMID 管理領域」を実現していることを、それぞれの背景に配置したライトグリーンの楕円で表現しています。なお、「UMID 管理領域」の詳細については改めて 5.3.3 節で紹介しますが、ここでは、それは当該システムの構成要素の各々が、4.3.2 節で紹介した UMID 管理機能を適切にサポートしていることと等価であるとお考え下さい。

さて、従来、このようなファイルベース制作・送出システムは、映像制作・送出のトータルソリューションとして、幾つかの業界ベンダから垂直統合型のまとまったシステム形で提供されてきました。この場合、当該システムの構成要素は基本的には当該ベンダ製品となりますから、そのベンダが独自で提供する素材 ID スキームをシステム全体に展開することで、システム内のシームレスな素材 ID 運用を実現しています。

一方、業界ベンダの各製品にはその目的に照らし合わせてそれぞれ強みや弱みといったこともありますので、

⁸ 厳密に言えば、5.3.2 節で紹介した「UMID 応用原理」を遵守した UMID の取り扱いも求められることとなります。

各々のシステム構成要素に対して複数のベンダ製品から最適なものを選択し、それらを組み合わせることでユーザのビジネス要件を最大限に満足できるシステムをユーザ自らが主導権をもって構築する、いわゆる「ベスト・オブ・ブリード」と呼ばれるシステム構築手段もあります。この場合、当該システムの構成要素は各々が独自の素材 ID スキームを保有している訳ですから、ここで解決すべき課題はまさに、4.3.2 節で議論した、独立した複数 MAMS の連携運用における共通素材 ID の問題と同じものであるといえます。

つまり、システム内共通素材 ID としての UMID の本領が発揮されるのは、まさにここで示した「ベスト・オブ・ブリード」なシステムのように、それぞれが独立主体として振る舞うものを構成要素としてシステム全体を構築する場合となります。

そして、当該システム構成要素としては、単一映像機器/アプリのみならず、小規模サブシステムや外部クラウドサービスの活用など任意の実装形態をとることが可能ですので、今後の当該システムの機能拡張や部分更新などの要求に対してより柔軟に対応していくことを所望するのであれば、業界標準映像素材識別子である UMID のシステム内共通素材 ID としての運用を考慮しておくことが、これからのファイルベース制作・送出システム構築における一つの重要な鍵になると考えられます。

4.4. Media SOA における UMID の応用

4.4.1. Media SOA とは？

4.3.3 節で紹介したベスト・オブ・ブリード的なシステムの構築ですが、以前からこれが理想とはされながらも、システムの構成要素間の接続とその検証に多くの開発工数を要することから敬遠されるくらいがありました。

しかし 2000 年代初頭に始まった Web サービスを用いた標準的なシステム連携技術の開発や、SOA (Service Oriented Architecture) の発想に基づくシステム基盤テクノロジーの確立により、ベスト・オブ・ブリード的なシステム構築はより身近なものとなり、特に、会計処理や販売管理などといったいわゆる基幹業務(エンタープライズ)システムの構築においては、既に有力な選択肢の一つとして幅広く提供されています。

そこでこの SOA の発想をファイルベース制作・送出システムの構築にも適用しようというのが Media SOA の考え方であり、これを推進する業界標準化活動として、FIMS (Framework for Interoperable Media Services) [20]なる取り組みが、欧米の放送局や業界ベンダを中心に活発に

展開されています。

以下、この Media SOA について簡単に紹介します。なお、Media SOA 及び FIMS のまとまった概要紹介については、文献[21]をご参照下さい。

図 11 は、従来の発想に基づくシステム構造と SOA の発想に基づくそれとを比較したものです。

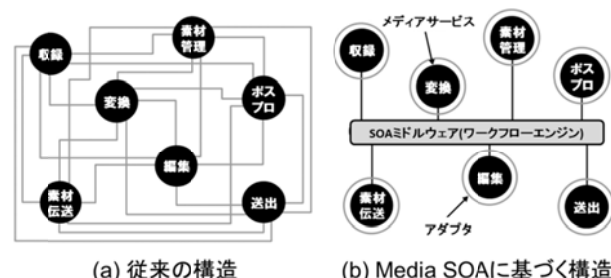


図 11 システム構造の比較

従来、ファイルベース制作・送出システムは、図 11 (a) に示すように、まずは所望の作業手順(ワークフロー)を定め、それに応じて必要な機材を調達し、それらを密に結合することで構築されてきました。従って、それが前提としたワークフローに則った作業においては高いコストパフォーマンスが得られたものの、ビジネス環境が大きく変化しワークフローの変更が求められても、それに柔軟に対応してシステム構成を変更することは必ずしも容易ではありませんでした。

そこで発想を大きく転換し、まずはシステムを特定のワークフローとは独立した汎用的なかたちで準備しておき、実際の作業に対しては、ワークフローが確定した段階でそれを記述したワークフロー記述データに基づきシステムを動的に構築するというのが Media SOA です。

すなわち Media SOA の発想では、図 11 (b) で示したように、対象とするシステムを、一般的なワークフロー分析に基づいて必要な機材を洗い出すものの各器材が提供する機能を「メディアサービス」として抽象化し、所望のメディアサービスを必要な時に所定のインターフェースを介して「オンデマンド」で呼び出すことができるよう、それらを「SOA ミドルウェア」に「疎結合」させた汎用システムとして構築しておきます。

そして実際の作業で用いるための具体的なシステムは、SOA ミドルウェア上にて「ワークフローエンジン」と呼ばれるアプリを実行し、入力されたワークフロー記述データに基づいて、必要とされるメディアサービスを順次呼び出して連携させること(これをオーケストレーションと呼びます)で実現されます。

これより、ワークフローエンジンを実行する際の入力ワークフロー記述データをビジネス環境の変化に応じて

変更するだけで、対象とするシステムのシステム構成を所望のかたちに簡単に変更することができる訳です。

なお、前出した FIMS が取り組んでいるのは、先述したメディアサービスのインタフェースの業界標準化です。本稿執筆時点においては、Capture、Transfer、Transform、Content Repository、Quality Analysis の5つのメディアサービスについて、標準化完了あるいは検討中の状況にあります[20]。

4.4.2. Media SOA システムにおける UMID の応用

では、このような Media SOA システムにおいて、UMID はどのような役割を果たし得るのでしょうか？ 図 12 はその説明のために、4.2.2 節の図 7 で紹介したシステムに対して SOA ミドルウェアや他のアプリを追加したものをベースに、UMID の役割を模式的に示したものです。

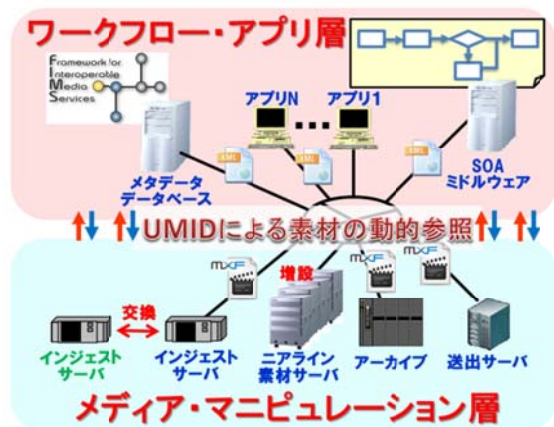


図 12 Media SOA システムにおける UMID の役割

4.2 節では、素材の内容を記述するメタデータが XML ファイルとして生成され、メタデータ・データベースに格納される様子を説明しましたが、実は、4.4.1 節で紹介したワークフロー記述データはもとより、各種メディアサービスとのインタフェースにおいて受け渡しされる制御関連データに至るまで、図 12 の上位の領域でやり取りされる殆どの全てのデータは XML で記述されたものであると言っても過言ではありません。そして、ここで取り扱われる XML ファイルは、そのデータサイズは大きくとも高々数百 KB 程度ながら膨大な種類があり、またそれらのデータ処理は、対象としたアプリの利用目的やワークフローの変更、更にはメディアサービスの違いなどに応じて柔軟に対応することが求められます。

一方、図 12 の下位の領域でやり取りされるのは、典型的には MXF ファイルとして生成、記録された素材の格納実体そのものですから、そのデータサイズは小さくとも数十 GB、場合によっては TB オーダに至るなど大規

模なものとなります。しかしながらその処理内容は、基本的には、データの移動、変換、蓄積といった比較的単純なものに過ぎません。

このように、各領域で取り扱われるデータとそれらの処理内容の性質が大きく異なることから、それぞれの領域を分けて取り扱うのが好適となります。そこで、当該上位領域を「ワークフロー・アプリ層」、同下位領域を「メディア・マニピュレーション層」と呼んで区別することにします。

ところで現状の Media SOA システムでは、ワークフロー・アプリ層とメディア・マニピュレーション層との間の連携に URL が用いられています。具体的には、ワークフロー・アプリ層に存在する XML ファイルが、それに関連付けられた MXF ファイルを、その URL を指定することで外部参照するという形式をとっています。

さて、4.4.1 節で紹介した FIMS が検討しているのは抽象化されたメディアサービスであって、図 12 によれば、これは専らワークフロー・アプリ層での議論となります。

この層は、状況の変化に柔軟に対応できることが特徴であるとは言え、一旦ワークフロー及びそこで用いられるメディアサービスが定まったならば、次にそれらに変更が入るまでは当然のことながら定常状態で動作します。

一方、実際の映像制作の現場においては、図 12 が示すように、例えば「インジェストサーバ」に不具合が発生したので急きょ、そのバックアップ品と交換したいとか、「ニアライン素材サーバ」から容量不足の警告があったので直ちにそれを増設したいといった、メディア・マニピュレーション層のみに閉じた範囲での対応も頻繁に求められます。

ところが現状のようにワークフロー・アプリ層とメディア・マニピュレーション層との間の連携に URL を用いている限り、メディア・マニピュレーション層のみに閉じた範囲での対応であっても、それを簡単に実施することはできません。なぜなら、ワークフロー・アプリ層で取り扱う XML ファイルに含まれた URL の“http://サーバ名/ファイルパス/ファイル名”なる表記ルールがサーバ名の明示的な指定を求めていることから、交換あるいは新たに増設された素材サーバなどそのサーバ名が不明の場合、それに格納された素材ファイルを外部参照することができないからです。

実はこの問題を解決するのも、業界標準映像素材識別子としての UMID の重要な役割の一つです。

すなわち 4.2.1 節の図 6 で紹介したように、XML ファイルから MXF ファイルへの外部参照を、当該 MXF ファ

イルのありかを示した URL の代わりにそれを一意に特定する UMID を用いて指定することで、当該 MXF ファイルを格納した素材サーバのサーバ名を省略することが可能となります。

もちろん UMID そのものには当該 MXF ファイルがどこにあるかといった情報を含んではいませんので、実際に当該 MXF ファイルへのアクセスが必要となれば、その時点で当該 MXF ファイルをアクセスする手段としての URL の入手が必要となります。

これはまさに、4.2.2 節で紹介した UMID に基づく素材検索において、所望の素材を一意に特定する UMID を知り得た映像系「アプリ」が、当該素材の格納の有無を素材サーバに問い合わせるやり取り(図 7 中の③及び④)に他なりません。すなわち、メディア・マニピュレーション層に配置された素材サーバなどのシステム構成要素がこのようなやり取りに能動的に対応できたならば、ワークフロー・アプリ層のアプリは、その必要性が発生した段階で、当該アプリが求める素材ファイルを一意に特定した UMID を、当該素材ファイルを直接アクセスする手段としての URL に変更するよう、各々の素材サーバに依頼すればよい訳です。

さらに素材サーバなどメディア・マニピュレーション層のシステム構成要素にとっても、自らの名前がワークフロー・アプリ層から明示的に指定される訳ではありませんので、当該システムの構成要素が交換あるいは新たに増設されたものであったとしても、それが格納すべき素材を必要に応じて他の素材サーバから適宜マイグレーションするなどといった作業を上位アプリから独立して実施することが可能となります。

このように、ワークフロー・アプリ層とメディア・マニピュレーション層との間の連携に対して現状の URL の代わりに UMID を用いることで、両者の独立性を飛躍的に高めることが可能となり、図 12 ではこれを、「UMID による素材の動的参照」と称しています。

別の見方をすれば、この UMID による素材の動的参照あるいはワークフロー・アプリ層とメディア・マニピュレーション層との間の動的な連携とは、まさに Media SOA の重要な特徴であるメディアサービス間の「疎結合」やメディアサービスの「オンデマンド」な呼び出しなどといった考え方を、当該システムで取り扱う素材に対しても適用したものであるといえます。すなわち、Media SOA システムにおける動的参照ツールとしての UMID は、Media SOA システムを、まさにその基本思想に則って更に進化させるための「鍵」になるということです。

4.5. 解決すべき課題

4.5.1. はじめに

本章では主に、大域的一意な素材識別子としての UMID の応用事例を紹介してきました。ここで、4.2.1 節で紹介した MXF ファイルとしての素材と XML で記述されたメタデータとの UMID を介した関連付けについては、既に XDCAM メタデータとして実用化済みであることは、3.4.2 節で紹介した通りです。

しかし、それ以外の UMID の応用事例については、実験レベルはともかく、商用ベースの実用化という意味では、実は今のところ皆無です。

その最大の原因は、このような UMID 応用を実用化する上で必要とされる UMID 応用に関する基本技術、すなわち「UMID 解決プロトコル」と「UMID 応用原理」がこれまで確立されていなかったことにあります。

以下、これらの基本技術について簡単に紹介します。

4.5.2. UMID 解決プロトコル

図 1 で示した UMID フォーマットから明らかのように、UMID それ自体には、それが大域的一意に特定する素材がどこにあるかといった情報までは含んでいません。

これより、実際に所望の素材にアクセスしたい場合には、当該素材を一意に特定する UMID から、ファイルとしての当該素材をアクセスする手段である URL を得る、いわゆる「UMID 解決」を実現する必要があります。

4.2.2 節で紹介した UMID に基づく素材検索において、図 7 中の③、④で示した映像系「アプリ」と「インジェストサーバ」とのメッセージのやり取りは、まさにこれを具現化した一例です。なお、図 7 では当該アプリは問い合わせメッセージを全ての素材サーバに一斉に送付していますが、当該アプリが各々の素材サーバに順次、問い合わせを実施し応答を得る形式でも構いません。

ただ、ここで大切なのが、映像系「アプリ」や各種の素材サーバは様々な業界ベンダから独立して提供されるでしょうから、そのような状況においても、任意のファイルベース機器/アプリ間において、先述した UMID 解決の為のメッセージのやり取りが確実に実施できるようにすることです。

これはすなわち、UMID 解決の為のメッセージのやり取り、つまり「UMID 解決プロトコル」(UMID Resolution Protocol)を、業界標準規格として開発、策定する必要があるということに他なりません。

なお、4.4.2 節の図 12 で議論した、UMID を介したワークフロー・アプリ層とメディア・マニピュレーション

層と間の「疎結合」の実現の鍵となるのもまた、ここで議論した UMID 解決プロトコルとなります。

UMID 解決プロトコルの詳細及び「SMPTE UMID 応用プロジェクト」におけるその取り組み状況については、5.4 節をご参照下さい。

4.5.3. UMID 応用原理

ここまで、UMID を、「素材を大域的一意に特定する素材識別子」と述べてきましたが、ここでいう素材の厳密な定義は何であって、UMID はそういった素材のどのような側面を大域的一意に特定しているのでしょうか？

また、2.3 節で紹介したように、UMID それ自体には、「大域的一意な素材識別子」の代わりに「関連付けツール」としての機能を提供することも期待されていますが、これらはどのような状況においてどう使い分けされるべきなのでしょう？

更に、3.3 節と 3.4 節のそれぞれにおいて、VTR とファイルベース機器における UMID 応用の実例を紹介しましたが、UMID 応用体系の全容を俯瞰した場合、それらはどう位置付けされるべきなのでしょう？

「UMID 応用原理」とは、このような質問への回答を論理的に導き出す際の「公理」に相当するものであり、換言すれば、UMID 応用に関わる全ての業務用映像機器及び映像系アプリが遵守すべき最も基本的な運用ルールを定めたものです。

今後、様々な用途において新たな UMID 応用が展開される可能性がある中で、あらゆる UMID 応用を開発する際の共通基盤としての「UMID 応用原理」を曖昧なく規定することは、相互運用性を保証した UMID 応用体系の全体が将来に渡って破たんをきたさないようにするための礎となります。

UMID 応用原理の詳細及び「SMPTE UMID 応用プロジェクト」におけるその取り組み状況については、5.3 節をご参照下さい。

4.5.4. MXF における UMID の応用

これは、本章で紹介した UMID 応用事例を実用化の上での直接の妨げとなった原因ではありませんが、UMID 応用検討において解決すべき重要課題の一つでありますので、ここで紹介させていただきます。

3.4.1 節では、「ある素材を大域的一意に特定する UMID が、当該素材を与える MXF ファイルのヘッダ部分に埋め込まれている。」と説明しましたが、実は一つの MXF ファイルには、図 1 の UMID フォーマットに則った三種

類の「メタデータ項目」が含まれています。もちろんその内の一つは、当該素材を大域的一意に特定する素材識別子としての UMID ですが、MXF 技術の仕様上、当該 UMID は必ずしもそれを含む MXF ファイルそのものを一意に特定する識別子と等価であるとは限りません。

従って、4.5.3 節で議論した「UMID 応用原理」が MXF 技術の文脈の中でどう展開するのかの分析を通じて、当該三種の「UMID 型メタデータ項目」のそれぞれが果たすべき役割を明確にする必要があります。

MXF における UMID の応用の詳細及び「SMPTE UMID 応用プロジェクト」におけるその取り組み状況については、5.5 節をご参照下さい。

5. SMPTE UMID 応用プロジェクト

5.1. はじめに

本章では、「SMPTE UMID 応用プロジェクト」の概要とこれまでの経緯、そして同プロジェクトで取り扱う UMID 関連技術の紹介並びにそれらの技術検討の現状について報告します。

5.2. SMPTE UMID 応用プロジェクト

5.2.1. SMPTE UMID 応用プロジェクトとは？

「SMPTE UMID 応用プロジェクト」とは、UMID 応用の実用化と更なる展開を目的に、筆者らが SMPTE 標準化コミュニティで始めた一連の活動の総称で、2012 年に最初に開始した「SMPTE 30MR 技術委員会 UMID 応用調査グループ」(SMPTE TC-30MR SG (Study Group) UMID Applications)[22]を母体に、必要に応じて様々な SMPTE 標準規格策定作業班を設置して活動を進めています。

同プロジェクトの議長は筆者が務め、本稿執筆時点において、38 の団体から、42 名のエキスパートにプロジェクトメンバーとして参加登録していただいております。

5.2.2. SMPTE UMID 応用プロジェクト発足までの経緯

そもそも本プロジェクトを提案する切っ掛けとなったのは、2011 年 10 月に開催された SMPTE 年次技術会議での筆者らの発表でした[23]。

この中で筆者らは、UMID が SMPTE 標準規格になって 10 年以上が経過したにも関わらず、事実上、全く使われていない状況をかんがみ、その主たる原因が、まさに 4.5 節で紹介した業界標準規格としての「UMID 解決プロトコル」と「UMID 応用原理」の欠如にあることを指摘しました。

この発表は当時、大きな話題となり、SMPTE 会議事務局から当日の筆者の発表の様子が YouTube に投稿されたり[24]、後日、SMPTE のメインホームページで大々的に取り上げられたり[25]、当該発表の予稿が SMPTE 機関誌に再掲されたりしました[26]。

これに背中を押され、2011 年 12 月に開催された SMPTE 標準化コミュニティ Burbank 会合に参加してそれらの SMPTE 標準規格を策定するための活動を提案し、2012 年 3 月の同 Atlanta 会合での審議を経て翌月に正式承認を受け、プロジェクトの発足に至りました[27]。

5.2.3. これまでの活動概要

当初、まずは「UMID 応用原理」の SMPTE 標準規格化を目的に RP 205[2]旧版の改定作業を提案した「SMPTE UMID 応用プロジェクト」でしたが、UMID は曲がりなりにも既に SMPTE 標準規格として確定しており、また第 3 章で紹介した筆者らの取り組みを始め、実験レベルも含めると、それまでも UMID 応用に関する様々なトライアルが実施され、報告されていました[28]。

そこで上位委員会(SMPTE TC-30MR)からは、プロジェクトの設置において、最初は SG (Study Group)として発足し、UMID 応用に関する現状調査から開始するよう指示が出されました。

これよりプロジェクト発足後、まずはそれまでに公表済みの UMID 応用に関する様々な文献をかき集めた他、2012 年 9 月の SMPTE 標準化コミュニティ Geneva 会合に合わせて「UMID 応用会議」[29]を開催し、関係者から UMID 応用に関する様々な取り組みについて紹介していただくなど、内外からの情報収集に努めました。

また、これと並行して内外に向けた「SMPTE UMID 応用プロジェクト」の広報活動を積極的に展開しました。

具体的には、2012 年 4 月の 2012 NAB Show [30]や、同年 6 月にシンガポールで開催された BroadcastAsia 2012 [31]などに参加して、当該プロジェクトの活動紹介をおこないました。

更に ABU (Asia-Pacific Broadcasting Union: アジア太平洋放送連合)から招待講演の依頼を受けて、その年の 10 月にソウルで開催された同技術委員会会合[32]に参加し、また翌 2013 年 10 月には再び SMPTE 年次技術会議[33]へ参加して、SMPTE 標準化コミュニティを超えた多くの業界エキスパートに対して、「SMPTE UMID 応用プロジェクト」の活動紹介並びに進捗報告を実施しました。

そして、このような活動を通じて収集した様々な UMID 応用事例を分析した結果、SMPTE 標準規格とし

て規定すべき UMID 応用に関する最も基本的な共通の運用ルールを抽出し、これを「UMID 応用原理」として、分析対象とした UMID 応用事例と共に「UMID 応用調査報告書パート 1」なる報告書[34]にまとめ、2013 年 2 月に上位委員会へ提出しました。

その後、筆者らは、当該報告書の勧告を踏まえて「SMPTE RP 205 改定作業班」の設置を提案し、上位委員会からの承認を受けて、当該作業班を直ちに発足させました[35]。

続いて、UMID 解決プロトコルを始め、UMID 応用の更なる展開を目指して新たに策定すべき SMPTE 標準規格をまとめた「UMID 応用調査報告書パート 2」の作成に着手し、「UMID 解決プロトコル」を含むその前半部分を「UMID 応用調査報告書パート 2-1」[36]としてとりまとめ、2014 年 3 月に上位委員会へ提出しました。

そして筆者らは、当該報告書の勧告を踏まえて「UMID 解決プロトコル策定作業班」の設置を提案し、上位委員会からの承認を受けて、2014 年 7 月に当該作業班を発足させました[37]。

また、「MXF における UMID 応用の調査報告書」[38]を含めたその後半部分を「UMID 応用調査報告書パート 2-2」[39]としてとりまとめ、2015 年 7 月に上位委員会へ提出しました。そして本稿執筆時点において、当該報告書は、上位委員会によるレビューの段階にあります。

5.3. UMID 応用原理

5.3.1. はじめに

本節では、解決すべき課題として 4.5.3 節で紹介した「UMID 応用原理」の概要と、これを規定した改定 SMPTE RP 205 2014 年版[2]を紹介します。

なお本節で紹介する内容は、筆者が 2014 年度の映像情報メディア未来賞/フロンティア賞を拝受した直接の理由に相当するものです。

5.3.2. UMID 応用原理の概要

4.5.3 節で述べたように、「UMID 応用原理」とは、UMID 応用に関わる全ての業務用映像機器及び映像系アプリが遵守すべき最も基本的な運用ルールを定めたものです。

UMID 応用原理は、後述する 7 項目から構成され、それぞれ基本ステートメントと、その意図するところを明確にするための補足説明文から構成されます。

以下、各項目について簡単に紹介します。

Principle 1 – Definitions

UMID 応用原理で用いられる語句を定義しています。

特にUMIDが識別対象とする「素材」の厳密な定義に加え、「オリジナル素材」(マテリアル)と、「派生素材」(インスタンス)とを明確に区別しています⁹。

Principle 2 – UMID Creation

オリジナル素材を生成した場合には、2.3.1節で示したように、新規生成したMat.#とゼロ値のInst.#で構成されるUMIDを当該素材に付与することを求めています。

Principle 3 – UMID Integrity

異なるオリジナル素材は異なるUMIDにて区別されなければならないことを求めています。

Principle 4 – UMID Identification

複数のオリジナル素材が一つのUMIDを共有した場合、当該全てのオリジナル素材は、それらの再生においてビット単位で同一であることを求めています。

これは換言すれば、UMIDが大域的一意に特定しているのが、素材を再生した時に得られるベースバンド・ビットストリームであることを意味しています。

従って、例えば異なるファイル書式で記録されるなど素材ファイルの格納実体としては異なるものであったとしても、それらの再生が完全に同一の結果をもたらすのであれば、それらはまとめて一つのUMIDで特定してもよい、ということになります。

Principle 5 – UMID Inheritance

オリジナル素材から派生素材を生成した場合には、2.3.2節で示したように、当該派生素材に対して、当該オリジナル素材から継承したMat.#と非ゼロ値のInst.#で構成されるUMIDを付与してもよいことを示しています。

Principle 6 – Extended UMID

ある素材を構成するマテリアルユニットに付与された拡張UMIDの基本UMID部は、原則として当該素材全体を大域的一意に特定するUMIDと一致させるべきであることを求めています。なお、マテリアルユニット及び拡張UMIDの詳細については、5.6.2節をご参照下さい。

Principle 7 – Source Pack

ソースパックを含む拡張UMIDが付与されたオリジナル素材から派生素材を生成し、Mat.#を継承した拡張UMIDを生成、付与した場合、原則として最初に生成、付与されたソースパックをそのまま継承すべきであることを求めています。なお、ソースパックの詳細については、5.6.2節をご参照下さい。

5.3.3. UMID 管理領域

5.3.2節で紹介したUMID応用原理は、UMIDそのものの取り扱いについては明確に定めてはいるものの、その具体的な実施方法については何ら規定していません。

これは、様々な特徴をもった業務用映像機器及び映像系アプリが存在する中で、ある特定の方法でUMID応用原理の実現を縛るべきではない、換言すれば、UMID応用原理を遵守する限りにおいて、それぞれの機器/アプリが、その特徴に応じて自由にそれらを実装すればよい、との考え方に基づくものです。

とは言え、5.3.2節で紹介したUMID応用原理の基本ステートメントだけではあまりにも抽象的です。そこで、特にUMIDの大域的一意な素材識別子としての機能に着目してUMID応用原理の具体的な実施例を示すべく、「UMID管理領域」なる概念を導入しました。

「UMID管理領域」は、UMID応用原理に準拠したUMID - 特に大域的一意な素材識別子としてのUMID - が付与された素材だけで構成される論理空間として定義されます。そしてこの定義に基づけば、UMID管理領域を構成する素材に付与されたUMIDは、UMID応用原理的に常に正しい(信頼できる)、換言すれば、その大域的一意性が常に保証されている、ということになります。

ところで、2.3.1節では、その生成において専用のID登録サーバを必要としないことがUMIDの際立った特長の一つである旨、説明しました。ただ、これは決して、UMIDを登録、管理する機能が一切不要であることを意味する訳ではありません。

事実、3.4.3節で議論したように、UMIDの大域的一意性を維持するためには、UMIDの適切なメンテナンス対応が必要となります。そしてこのことは、所定の論理空間に存在する素材の様々な取り扱いにおいて、当該素材とそれに付与されたUMIDとを常に管理し、適切なメンテナンス対応を施す機能の導入が必要不可欠であることを示唆しています。

図13は、このことを踏まえて、UMID管理領域を模式的に示したものです。

図13では、UMIDが付与された様々な素材が、ある記録媒体の内外に保存されていること、また素材とそれに付与されたUMIDとを管理する役割を担う「素材管理部」が存在することが示されています。ここで素材管理部から幾つかの素材への赤破線の矢印は、当該素材管理部がそれらの素材の存在を認識し、管理している様子を表現しています。

⁹ ここでのマテリアルとインスタンスの違いは、UMID特有のものです。

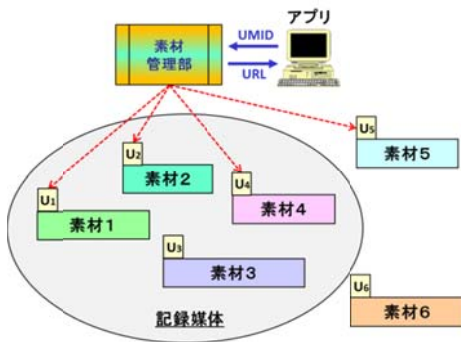


図 13 UMID 管理領域

そして図 13 において、「UMID 管理領域」は、当該素材管理部が管理対象とした、「素材 1」、「素材 2」、「素材 4」及び「素材 5」で構成される論理空間として示されます。換言すれば、図 13 の「素材 3」のように、UMID 管理領域を構成する素材と同じ記録媒体に保存されていてもそれが素材管理部の管理対象外であれば、当該素材は UMID 管理領域の構成要素ではなく、逆に「素材 5」のように、当該記録媒体の外部に存在していてもそれが素材管理部の管理対象であれば、当該素材は UMID 管理領域の構成要素となります。

また、図 13 では、その素材管理部が、外部の映像系「アプリ」から UMID を入力され URL を戻す様子を示していますが、これは 4.2.2 節の図 7 中③及び④で示した素材サーバの応答動作に相当するものです。すなわち素材管理部は素材及びそれに付与された UMID を管理していますから、これに対して例えば「U₁」なる UMID を与えたならば、当該素材管理部はそれが一意に特定する素材、例えば「U₁」が付与された「素材 1」へのアクセス手段としての URL を当該アプリへ戻すこととなります¹⁰。

5.3.4. UMID 管理領域のシームレスな結合

5.3.3 節で紹介した UMID 管理領域の概念は、複数の業務用映像機器あるいは映像系アプリが連携して素材管理をおこなう場合に特に重要になってきます。なぜなら、各々の機器/アプリがそれぞれの素材管理部によって UMID 管理領域を適切に実現しているならば、当該機器/アプリを連携させることで各々が提供する UMID 管理領域がシームレスに結合され、当該機器/アプリの全体をカバーする UMID 管理領域が自ずから実現されるからです。

以下、この UMID 管理領域のシームレスな結合について説明します。

いま、UMID 管理領域を実現したあるアプリ(アプリ 1)

¹⁰ もし入力 UMID が「U₃」であったならば、当該素材管理部は「知らない」と応答することになります。

が、新たな素材を自らの UMID 管理領域にインポートしたいとします。この時、仮に当該素材に予め UMID が付与されていたとしても一般的にはその大域的一意性は信頼できませんから、アプリ 1 の素材管理部が取るべき対応としては、当該素材のインポートの際に信頼できる UMID を自ら新規生成し、元々付与されていた UMID をそれで置き換えることで UMID の大域的一意性を担保する、ということになります。

一方、もしアプリ 1 が、インポート予定の素材が別のアプリ(アプリ 2)の UMID 管理領域にあったことを事前に知っていたならば、当該素材に付与された UMID は既にその大域的一意性が保証されている訳ですから、当該素材を丸ごと自らの UMID 管理領域にインポートする限りにおいて、アプリ 1 は当該素材に付与された UMID を信頼してそのまま再利用することが可能です。

そして、仮に当該インポート後にアプリ 2 に存在していた元素材を削除したならば、これはまさに、当該素材及びそれに付与された UMID の管理権が、アプリ 2 からアプリ 1 に「そのまま」委譲されたことと等価です¹¹。

更にいま、アプリ 1 とアプリ 2 の素材管理部が、例えば一方に問い合わせのあった UMID で特定される素材がその UMID 管理領域内に存在しない場合に当該問い合わせを他方にそのまま転送できたならば、両者が提供する UMID 管理領域は一つのまとまった UMID 管理領域とみなすことが可能となり、この場合はそもそもアプリ 2 からアプリ 1 への素材のインポートすら不要となります。

4.3.3 節で紹介したシステム内共通素材 ID としての UMID は、まさにこのような UMID 管理領域のシームレスな結合をシステム全体に渡って適用した結果、実現されるものです。そして 4.3.3 節の図 10 では、これを、システムの各々の構成要素の背景に配置したライトグリーンの楕円(UMID 管理領域)が互いにオーバーラップして一つの UMID 管理領域を構成し、それがシステム全体をカバーしている様子にて表現しています。

ここで改めて、2.3.1 節で紹介した ISAN や EIDR など、その生成、管理に専用の ID 登録サーバを必要とする素材識別子と UMID とを比較してみますと、前者は全ての素材を一元管理するための単一の ID 登録サーバが準備され、これが言わばトップダウン・アプローチで系全体の素材管理を実現しているのに対し、後者は素材識別子

¹¹ アプリ 2 の元素材を削除しなかった場合、UMID 応用原理的に同一の素材がアプリ 1 とアプリ 2 の両方で管理されることとなりますが、UMID を検索キーに当該素材を所望する外部アプリにとってはいずれを入手してもよく、両者の区別は不要であるといえます。

の生成、管理はあくまで局所的に実施されるものの、それらがシームレスに結合されることで、言わばボトムアップ・アプローチで系全体の素材管理を実現しているとみなすことができます。

特にこのボトムアップ・アプローチですが、もしその素材識別子の局所的な生成、管理の各々が完全に独立したものであれば、それぞれの機器/アプリの管理領域を超えて素材を移動させる場合には常に新たな素材識別子への付け直しが必要となりますから、これは実質的には、4.3.2 節の図 8 で紹介した、それぞれの機器/アプリが独自の素材 ID スキームを保有している場合と等価です。

しかし UMID の場合、UMID 応用原理が素材の「同一」性を定義し、当該「同一」素材が各々の UMID 管理領域を超えて移動する際に当該素材に付与された UMID の再利用を許していることが、UMID 管理領域のシームレスな結合、すなわちボトムアップ・アプローチによる系全体の素材管理の実現につながっているということです。

5.3.5. XDCAM™における UMID 管理領域の実現

さて、3.4.3 節では、XDCAM™シリーズが管理する素材は、それに付与された UMID によって大域的一意に特定できることを保証されている旨、紹介しました。これはまさに、XDCAM™シリーズが、5.3.3 節で定義した UMID 管理領域を実現していることに他なりません。

事実 XDCAM™では、例えばその代表的な記録媒体である光ディスクあるいは内部ストレージの単位において、そこに記録された素材を、それに付与された UMID と共に適切に管理しています。すなわち XDCAM™機器の各々は、記録媒体の単位において、UMID 管理領域を実現している訳です。

そしていま、あるファイルベース制作・送出システムが XDCAM™シリーズの製品群のみで構成されたならば、5.3.4 節で議論したように、各々の XDCAM™機器が記録媒体の単位で提供する UMID 管理領域をシームレスに結合することで、当該システム全体として一つの UMID 管理領域を構成できる、換言すれば、当該システム内で取り扱う任意の素材に付与された UMID は、その大域的一意性が保証されたものとなる、ということになります。

そして、当該システムの外部にある MXF ファイルを例えば FTP などを用いてこの UMID 管理領域に取り込む場合は、3.4.3 節で述べたように、原則として当該 MXF ファイルに元々付与されていた UMID を自ら新規生成したものに置き換えて取り込む訳ですが、取り込むべき MXF ファイルが当該システム内、すなわち別の

XDCAM™が提供する UMID 管理領域にあることを予め知っていた場合は、当該 MXF ファイルを丸ごと取り込むならば、それに付与されていた UMID もそのまま再利用するといったことも可能にしています¹²。

ただこのような対応は、XDCAM™シリーズといったある特定の業界ベンダの製品群で閉じた世界だからこそ実現できた訳であって、もしこれを 4.3.3 節で紹介したような「ベスト・オブ・ブリード」なシステムで実現したい場合は、異なる業界ベンダから提供される当該システム構成要素の各々が遵守すべき UMID の基本運用ルールを明確に定め、これを、強制力を有する業界標準規格にする必要があります。

これこそがまさに、「UMID 応用原理」の確立と、その SMPTE 標準規格化に相当する訳です。

5.3.6. 改定 SMPTE RP 205

“Application of Unique Material Identifiers in Production and Broadcast Environments”なるタイトルをもつ SMPTE RP 205 の初期版が発行されたのは、SMPTE ST 330 [1] と同時の 2000 年のことです。ただ、SMPTE ST 330 が UMID フォーマットやその各フィールド値の生成方法などを規定していたのに対し、SMPTE RP 205 は単に、それまでの検討の中で想定されていた様々な UMID 応用を抽象的に紹介しているに過ぎませんでした。

筆者らが 3.3 節で紹介した HDCAM™における UMID 機能の実装に取り組んだ際に参照したのが、まさにこれらの SMPTE 標準規格書でした。そして当該実装を通じて、当該規格書に様々な記載不備があったこと、また VTR における「素材」の定義や、更には 2.3 節で紹介したように、UMID には排他的な二つの機能があり、これらを明確に分離して取り扱う必要があることなどを見出しました。そこで、これらの知見を元に SMPTE 標準化コミュニティへ SMPTE RP 205 の改定作業を提案し、その結果発行されたのが同 2004 年版でした¹³。

このように、SMPTE RP 205 が元々その目的としていたのが、UMID を実際に運用する際の注意事項や参考情報の提供ということでした。そこで今回、「SMPTE UMID 応用プロジェクト」を通じて見出した「UMID 応用原理」を規定すべき文書としても、新たな SMPTE 標準規格書を起こすよりは、原点に立ち戻り、SMPTE RP 205 の改

¹² XDCAM™の FTP 仕様書では、これを UMID unchange mode による FTP Get と呼びます。

¹³ 同時に、GPS 情報の取り扱い不備などを改善した SMPTE ST 330 の 2004 年版が発行されました。

定として実施すべきとの判断に至りました。

これより改定 SMPTE RP 205 の記載内容は、旧版の記載事項を踏襲はするものの文書構成などは大きく変更し、事実上、スクラッチで書き換えをおこないました。その結果、ページ数も初版の 5 ページから 44 ページへと、大幅に増加することになりました。

今回改定した SMPTE RP 205 2014 年版[2]の記載内容は以下の通りです。まずは本文ですが、本規格書の適用範囲など所定の記載事項に続いて

UMID 応用原理(UMID Application Principles)

5.3.2 節で紹介した UMID 応用原理を規定しました。

UMID 対応アプリの考慮事項(UMID-Aware Application Considerations)

5.3.3 節で紹介した UMID 管理領域を始め、収録中の素材ファイルなど成長中素材(Growing material)における UMID の取り扱いルールなどを規定しました。

を記載しました。

なお、実は SMPTE RP 205 2014 年版の本文自体は、高々 10 ページ強に過ぎません。そして残りは付録(Annex)として、筆者らが「SMPTE UMID 応用プロジェクト」の活動を通じて見出した UMID 応用に関する様々な参考情報を報告しました。

以下、付録の記載内容ですが、

UMID の基本(UMID Basics)

2.2 節で紹介した基本 UMID 及び 5.6.2 節で紹介した拡張 UMID のフォーマット仕様を概要紹介しました。

UMID 管理領域の維持方法(To Maintain the UMID Managed Domain)

ファイルとしての素材の様々な取り扱いにおいて、UMID 管理領域を維持するために求められる一連の UMID メンテナンス対応方法の具体例を紹介しました。

その多くは、3.4 節で紹介した、筆者の XDCAM™ での UMID 機能の実装経験に基づくものです。

UMID の応用事例(UMID Application Examples)

第 4 章で紹介した UMID 応用事例を始め、UMID 応用原理の検討の際に収集した様々な UMID 応用事例について、それらの概要を紹介しました。

よくある質問 (Frequently Asked Questions)

「UMID 応用原理」、「UMID 管理領域」の概要説明を含め、これまでの検討の中で出された典型的な質問とその回答をまとめました。

特に、同じ素材識別子なるカテゴリに分類されることから、UMID と、2.3.1 節で紹介した ISAN や EIDR のどちらを採用すべきかといった議論が散見されま

したので、両者は排他的ではなく両立できるものであること、また 5.3.4 節で議論した素材管理に関するアプローチの違いから、映像制作現場での素材管理には前者が適しているのに対し、コンテンツ配信など完パケ管理には後者が適しているであろうことを、EIDR のエキスパートとも相談しつつ回答にまとめました。

から構成されます。

改定 SMPTE RP 205 は、2013 年 10 月にその初稿が上程され、SMPTE が定める標準規格策定プロセスを経て、2014 年 12 月末に正式に発行されるに至りました。

5.4. UMID 解決プロトコル

5.4.1. はじめに

本節では、解決すべき課題として 4.5.2 節で紹介した「UMID 解決プロトコル」の概要と、「SMPTE UMID 応用プロジェクト」におけるその検討状況を報告します。

なお本稿執筆時点において、UMID 解決プロトコルはまだ SMPTE 標準規格化が完了していませんので、本節では、「UMID 応用調査報告書パート 2-1」[36]で報告した内容を中心に紹介します。

5.4.2. UMID 解決プロトコルの概要

4.5.2 節で述べたように、「UMID 解決プロトコル」とは、所望の素材を一意に特定する UMID から、ファイルとしての当該素材をアクセスする手段である URL を得る、いわゆる「UMID 解決」を実現するためのやり取りのこととして、具体的には、4.2.2 節の図 7 中③及び④で示した映像系「アプリ」と「インジェストサーバ」とのメッセージのやり取りに相当します。

ただ実用面を考慮すると、URL 取得後の所望の素材ファイルの入手に際し、そのファイル書式や圧縮符号化方式といった当該素材ファイルに関する技術情報が不明なままだと、せっかく当該素材ファイルを素材サーバから手元にダウンロードしても結局は使えずといった事態にもなりかねず、これはそのデータサイズが数百 GB を超えることも珍しくない素材ファイルの入手手段としては致命的であると言わざるを得ません。

そこで、UMID 解決プロトコルにおける素材サーバの応答としては、単に入力 UMID で特定される素材ファイルの URL のみならず、クライアント側で所望の素材ファイルが利用可能かどうかを前もって判断できるような当該素材ファイルに関する技術情報も併せて返すことが求められます。

さらに 5.3.2 節で紹介したように、UMID 応用原理によ

れば、ある UMID が大域的一意に特定しているのは厳密には素材の再生時のベースバンド・ビットストリームです。すなわち、一つの UMID が一意に特定した素材に関し、その格納実体である素材ファイルとしては、例えばファイル書式(特に MXF、MP4 (MPEG-4 Part 14)、AVI (Audio Video Interleave)などのいわゆるファイルラッパー書式)や圧縮符号化方式(特に ZIP やロスレス JPEG など可逆圧縮符号化方式)などが異なる複数の素材ファイルが存在する可能性もあります。

これより UMID 解決プロトコルの素材サーバ側の応答が含むべき(UMID で指定された)素材に関する技術情報には、その再生時間長やピクセル当たりのビット深さ、色空間サブサンプリング方式など、当該素材の再生時のベースバンド・ビットストリームに関する技術的特徴を記述し、入力 UMID と 1:1 対応する項目群(これを「カテゴリー 1」と呼びます)と、ファイルラッパー書式や圧縮符号化方式など、当該素材の格納実体である素材ファイルに関する技術的特徴を記述し、入力 UMID と 1:n の関係にある項目群(同、「カテゴリー 2」といった二種類の項目群があることとなります)から、素材サーバは、この二つを明確に区別して応答する必要があります。

なお、冒頭で述べた所望の素材ファイルにアクセスする手段である URL ですが、これはまさに当該素材の格納実体である素材ファイルに関する技術的特徴の一つです。先述した「カテゴリー 2」の一項目として取り扱われることとなります。

5.4.3. DNS とサービス発見

ところで、本稿執筆時点において、UMID 解決プロトコルを実現するための基盤技術として最有力であるのは、インターネットの最重要インフラ技術の一つとして知られる DNS (Domain Name System)[40]です。

DNS は、一般には、与えられたホスト名をそれに割り当てられた IP (Internet Protocol) アドレスに変換するいわゆる「名前解決」をおこなう技術として知られていますが、実はその実体は、DNS RR (DNS Resource Record) と呼ばれる情報登録要素に基づく階層化分散型の汎用データベースに他なりません。

例えば、上述したホスト名から IP アドレスへの変換は、基本的には、当該ホスト名と IP アドレスとの対応を記述した「アドレス RR」と呼ばれる DNS RR を、名前解決を担う DNS ネームサーバに予め登録し、ホスト名を DNS クエリとして受けた当該 DNS ネームサーバが、それに

対応する IP アドレスを応答することで実現しています¹⁴。

そこで、このホスト名と IP アドレスとの対応の替わりに他の対応関係を DNS ネームサーバに登録することで、これまでも DNS を基盤とした様々なアプリが実現されてきました。

その中で最もよく知られているのが、そのような DNS の特徴を活用してネットワーク機器間の接続を容易にした、いわゆる Zeroconf と呼ばれる技術です。そこで以下、この Zeroconf の概要を紹介します。

Zeroconf とは、所定の閉じたネットワーク空間において DNS クエリを一斉配信するためのマルチキャスト DNS [41]と、DNS の「ポインタ RR」、「サービス RR」及び「テキスト RR」を活用した DNS-SD (DNS-based Service Discovery: DNS に基づくサービス発見) [42]とを用いて、当該ネットワーク空間に存在する所望の固有サービスの発見と、それへの自動的な接続を実現した技術です。

図 14 に、Zeroconf におけるサービス発見の様子を模式的に示します。

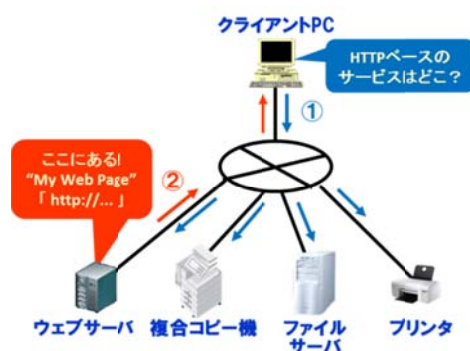


図 14 Zeroconf におけるサービス発見

図 14 において、「クライアント PC」は、HTTP ベースの所望の固有サービス(例えば「My Web Page」)へのアクセスを実現すべく、まずは HTTP ベースの任意のサービスを提供する全てのホストの所在を求める DNS クエリを、マルチキャスト DNS を用いて当該ネットワーク内に一斉配信します(図 14 中①)。

これを受けた「ウェブサーバ」は、自らが HTTP ベースの固有サービスである「My Web Page」を提供していることを、その固有サービス名とホスト名、そしてアクセスポイントなど接続パラメータと共に、「クライアント PC」に応答します(図 14 中②)。

この応答を受けた「クライアント PC」は、当該固有サービス名やホスト名を参考に、それが所望の固有サービスであるか否かを判断し、そうであれば同ホスト名の名

¹⁴ ここでは簡単の為、階層化ドメイン表記に基づく DNS ネームサーバの再帰的な名前解決処理は省略します。

前解決を通じて当該ホストに接続し、併せて入手した接続パラメータを参考にして当該固有サービスへのアクセスを実現することになります。

さて、図 14 によれば、ネットワークに接続された各種のサービス提供ホストには、それぞれが独自の DNS ネームサーバとして振る舞い、受信した DNS クエリに応答することが求められます。そしてこの応答メッセージのフォーマットなどは、当該 DNS ネームサーバをあたかも汎用 DNS ネームサーバであるとみなし、これに所定の DNS RR を登録した場合に出力されるであろう応答メッセージのそれらと等価です。

具体的には、DNS-SD の場合、各サービス提供ホストが出力する応答メッセージは、「ポインタ RR」にて HTTP ベースのサービスといったサービス種と「My Web Page」といった固有サービス名との対応を、「サービス RR」にて当該固有サービス名とそれを提供するホスト名との対応を、そして「テキスト RR」にて当該固有サービス名とそれにアクセスするための接続パラメータなどとの対応を記述し、それらの DNS RR を汎用 DNS ネームサーバに登録した場合に出力されるであろう応答メッセージと等価となります。

そして、DNS クエリを受信しそれに応答可能な全てのサービス提供ホストがそれらの DNS RR 登録に基づく応答メッセージを「クライアント PC」に戻すことで、DNS に基づくサービス発見を実現している訳です。

5.4.4. DNS に基づく UMID 解決プロトコルの実現

4.2.2 節の 図 7 と 5.4.3 節の図 14 との比較から、DNS クエリとしてサービス種の代わりに UMID を、戻り値として固有サービス名などの代わりに 5.4.2 節で議論した素材に関する技術情報項目群を与えることで、Zeroconf の代わりに DNS に基づく UMID 解決プロトコルが実現できることが判ります。

尤も Zeroconf の場合には 3 種類の DNS RR を必要としましたが、UMID 解決プロトコルの場合は所望の素材ファイルへのアクセス手段として応答される URL に当該素材ファイルを格納した素材サーバ名が含まれることから、単に「テキスト RR」を用いて、UMID と、それが一意に特定する素材に関する技術情報項目群との対応を記述するだけで十分です。

図 15 は、そのような考えを踏まえ、DNS に基づく UMID 解決プロトコルデモの一例として、BIND と呼ばれる汎用 DNS ネームサーバに後述するテキスト RR を予め登録したもの(これは「UMID 解決サーバ」に相当しま

す)と、「nslookup」なるコマンドライン型の DNS クライアントとを用いて、UMID 解決の様子をシミュレーションしたものです。

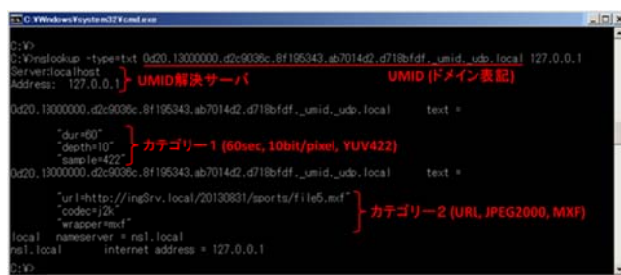


図 15 DNS に基づく UMID 解決プロトコルデモの一例

図 15 では、“06_h0A_h 2B_h 34_h 01_h 01_h 01_h 05_h 01_h 01_h 0D_h 20_h 13_h 00_h 00_h 00_h D2_h C9_h 03_h 6C_h 8F_h 19_h 53_h 43_h AB_h 70_h 14_h D2_h D7_h 18_h BF_h DF_h”なる UMID 値を入力として、それが一意に特定する素材に関する技術情報項目群を求めた結果、「カテゴリ 1」の項目群として、当該素材の再生時の再生時間長(“dur=60”秒)、ピクセル当たりのビット深さ(“depth=10”ビット)、YUV 色空間サブサンプリング方式(“sample=422”)が、また「カテゴリ 2」の項目群として、当該素材ファイルのファイルラッパー書式(“wrapper=MXF”)、圧縮符号化方式(“codec=j2k”)、URL (“url=http://ingSrv.local/20130831/sports/file5.mxf”)が戻された結果が示されています。

ここで「nslookup」の引数ですが、第一引数はそれが「テキスト RR」なる DNS RR で登録された情報への検索であることを、第二引数は後述するドメイン表記の UMID 値を、また第三引数は「UMID 解決サーバ」(この場合は、バックグラウンドプロセスとして実行中の BIND)の IP アドレスを示しています。

そして先述したドメイン表記の UMID 値ですが、第二引数を再掲すると以下ようになります。

```
0d20.1300...bfdf._umid._udp.local
```

ここで左端の UMID 値¹⁵に続く“_umid._udp.local”なるドメイン名は、テキスト RR の要求仕様及び DNS-SD での慣習に従い、当該 UMID が所定の閉じられたネットワーク空間における「UMID 解決」なる UDP ベースのサービスを求めたものであることを表しています。

また、図 15 のデモでは、汎用 DNS ネームサーバである BIND に対して

「カテゴリ 1」項目群

```
0d20.1300...bfdf._umid._udp.local IN TXT
("dur=60" "depth=10" "sample=422")
```

¹⁵ この UMID 値は、固定値である先頭 10byte を省略した結果、11 バイト目(0D_h)から始まっています。

「カテゴリ 2」項目群

```
0d20.1300..bfd.f._umid._udp.local IN TXT
  ("url=http://.../file5.mxf" "codec=j2k"
  "wrapper=mxf")
```

なるテキスト RR 記述を予め登録したものを「UMID 解決サーバ」に見立て、そこからの応答メッセージを受信した「nslookup」が、それらをそのまま表示しています。

なお 5.4.2 節で述べたように、「カテゴリ 2」については一つ以上の項目群が登録される可能性もありますが、ここでは簡単のため、一つの項目群だけが登録されていることにご注意下さい。

もちろん 4.2.2 節の図 7 の場合に、各々の素材サーバが実際に汎用 DNS ネームサーバの機能を持つ必要はありませんが、少なくとも各素材サーバには、それを提供する業界ベンダに関わらず、例えば図 15 で示したような「nslookup」とのやり取りの実現が求められます。

そしてこれを達成する上で必要なのが、まさに DNS に基づく UMID 解決プロトコル仕様を SMPTE 標準規格として策定するという他にありません。

5.4.5. UMID 解決プロトコル策定作業班

5.4.1 節で述べたように、本節で紹介した UMID 解決プロトコルの説明は、2014 年 3 月に上位委員会へ提出した「UMID 応用調査報告書パート 2-1」[36]で報告した内容に基づくものです。

「SMPTE UMID 応用プロジェクト」としては、その報告に基づいて直ちに「UMID 解決プロトコル策定作業班」[37]の設置を提案し、2014 年 7 月より当該作業班としての活動を始め、2014 年 12 月に UMID 解決プロトコルに関する SMPTE 標準規格書の初稿を作成したところです。

以下、本稿執筆時点で構想中の、当該 SMPTE 標準規格書の基本構成について、簡単に紹介します。

実は ICT 分野において、「名前解決」はまさに古くて新しい課題であり、DNS 以外にも Handle System [43]など様々なツールが提案されてきました。

更に 4.4.1 節で紹介した Web サービスに基づくメディアサービス・インタフェースの標準化などの動向を踏まえると、UMID 解決プロトコルの実現方法として DNS は引き続き最右翼ではありつつも、それだけに限定された仕様はあまり適切ではないとの判断に至っております。

そこで「UMID 解決プロトコル」仕様書としては、まずは特定の実装に依存しない概念モデルとしてこれを規定し、次にその具体的な実装方法として、例えば DNS に基づく UMID 解決プロトコルを実現する上で汎用

DNS 仕様に追加すべき項目を SMPTE 標準規格として規定するというアプローチを検討しています。

UMID 解決プロトコルに関する SMPTE 標準規格書は、2016 年度中の完成、発行を目標にしています。

5.5. MXF における UMID の応用

5.5.1. はじめに

本節では、解決すべき課題として 4.5.4 節で紹介した「MXF における UMID の応用」について、「SMPTE UMID 応用プロジェクト」における検討結果の概要を紹介いたします。

なお本稿執筆時点において、MXF における UMID の応用に関しては、その検討結果をまとめた「MXF における UMID 応用の調査報告書」[38]を含む「UMID 応用調査報告書パート 2-2」[39]が上位委員会に提出され、レビューの段階にあります。

そこで本節では、この「MXF における UMID 応用の調査報告書」で報告した内容を中心に紹介します。

5.5.2. MXF ファイルの概要

最初に MXF ファイルの概要を紹介いたします。なお、ここで紹介する MXF ファイルの説明は、本節での MXF における UMID の応用の議論を理解していただく上で必要最小限なものに留めています。MXF 自体には多彩な機能がありますので、MXF についてより詳しい情報を所望される場合は、例えば文献[44]などをご参照下さい。

図 16 に、MXF ファイルの構造を模式的に示します。他の多くのメディアファイルと同様、MXF ファイルもまた、様々なメタデータを格納した「ファイルヘッダ」、映像音声データ(以降、「エッセンスデータ」と呼びます)を格納した「ファイルボディ」、ファイルの末尾を示し、必要に応じてメタデータを格納した「ファイルフッタ」から構成されます。



図 16 MXF ファイルの構造

ここでファイルヘッダには、MXF ファイルとして与えられた素材のタイトルや内容メモといったいわゆる「内容記述メタデータ」と、ファイルボディに含まれるエッセンスの再生方法などを記述した「構造メタデータ」が、論理的なツリー構造を構成するかたちで含まれています。またファイルボディには、「汎用コンテンツ」[45]を介し

て、例えばフレームといった編集の最小単位(これを「エジットユニット」と呼びます)で区分けされたエッセンスデータが格納されています。

ここで特筆すべきは、汎用コンテナ自体には、それが格納するエッセンスデータの再生方法といった情報は一切含まれておらず、当該情報を記述するのは、専らファイルヘッダに含まれる構造メタデータの役割であるということです。それ故、全ての MXF ファイルにとって、構造メタデータは必須のメタデータとなっています。

このような構造の結果、MXF ファイルは、他のメディアファイルには見られない興味深い機能を提供しています。それは、ある MXF ファイルの再生において、それに含まれるエッセンスデータを部分的に指定し、当該指定部分だけを実際の再生対象として外部へ出力することができるという機能です¹⁶。

図 16 ではこれを、ファイルボディ内の汎用コンテナの一部分が In/Out 点で指定され、再生時にはその部分だけが外部へ出力される様子にて示しています。

そして、このような機能を実現しているのが、MXF ファイルのボディに含まれる例えば各フレームの時間軸上の配置情報などエッセンスデータの時間軸を記述した「MXF FP」(MXF File Package: MXF ファイル・パッケージ)と、その再生時に実際に外部へ出力されるエッセンスデータの時間軸を記述した「MXF MP」(MXF Material Package: MXF マテリアル・パッケージ)と呼ばれるメタデータ項目です。

5.5.3. MXF ファイルの内部動作モデルと UMID

では、MXF ファイルの再生時、MXF MP 及び MXF FP は、具体的にはどのように振る舞うのでしょうか？これを模式的に表したのが、図 17 となります。

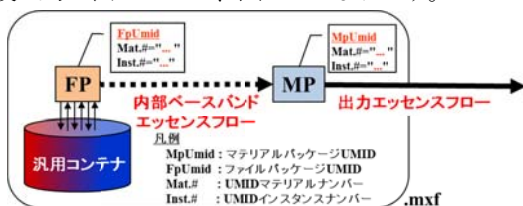


図 17 MXF の内部動作モデル

図 17 は、ある MXF ファイルを再生した際、MXF FP が、汎用コンテナに含まれる全てのエッセンスデータそのまま読み出して再生することで当該 MXF ファイル内部でのエッセンスフローを生成する様子を、また MXF MP が、当該内部エッセンスフローを受けて所定の制御

をおこなうことで実際に外部へ出力するエッセンスフローを生成する様子を示しています。

ここで、汎用コンテナに含まれるエッセンスデータは通常は圧縮符号化されたかたちで格納されていますので、これをベースバンド信号に伸張復号化するのはまさに MXF FP の役割です。このため MXF FP には、当該圧縮符号化方式に関する様々な技術パラメータなどが記述されています。

一方、MXF MP は、MXF FP からベースバンド信号として受信した内部エッセンスフローから実際に外部へ出力する部分だけを切り出すといった、いわゆるスイッチャー的な役割を果たしています。すなわち図 16 において、汎用コンテナに対して In/Out 点を指定していたのは、まさにこの MXF MP であったということです。

そして MXF ファイルにおける UMID は、これら MXF MP 及び MXF FP を一意に特定するための識別子として付与されており、それぞれ、「MpUmid」、「FpUmid」と呼びます。

これらは、元来はそれぞれのメタデータ項目の実体データ(インスタンス)を一意に特定するための識別子ですが、先述した MXF MP 及び MXF FP の動作を考慮すれば、MpUmid 及び FpUmid はまた、それぞれが識別する MXF MP 及び MXF FP が生成するエッセンスフロー自体を一意に特定する識別子であるとみなすことも可能です。

そのように考えると、ある MXF ファイルを再生した時に実際に外部へ出力されるエッセンスフローを一意に特定しているのが MpUmid となりますから、5.3.2 節で紹介した UMID 応用原理の Principle 4 (UMID Identification) に照らし合わせれば、3.4.1 節で触れた MXF ファイルとしての素材を大域的一意に特定するためにそのヘッダ部分に埋め込まれていた UMID とは、まさにこの MpUmid のことであったということです。

5.5.4. MXF ファイルにおける UMID の応用

MpUmid の役割

5.5.3 節では、MXF ファイルとしての素材を一意に特定しているのが、その再生時に外部出力されるエッセンスフローを記述した MXF MP に付与される MpUmid であると述べました。このことから、MXF ファイルとしての素材を対象とした UMID に基づく素材管理において、2.3.1 節で紹介した大域的一意な素材識別子として運用され、5.3.3 節で議論した UMID 管理領域を構成すべくメンテナンス対応の対象となり、そして 5.4 節で紹介した UMID 解決プロトコルの入力値となるのは、まさにこの

¹⁶ 最も広く使われている MXF OP1a とは、MXF ファイルの再生時において、内部に含まれるエッセンスデータの全てを再生対象として外部出力する形式のことです。

MpUmid であるといえます。

なお、先述したように MXF MP には、その再生時に外部出力すべきエッセンスデータを部分指定する機能がありますが、当該部分をわずかでも変更することは、外部エッセンスフローとして一意に特定される素材そのものを変更することと等価です。従ってこの場合、3.4.3 節で議論した EditList と同様、原則として元の MpUmid を新規生成した UMID 値へ置き換えることが求められます。

また 4.5.4 節でも述べましたが、MpUmid は多くの場合、それを含む MXF ファイルそのものを大域的一意に特定した識別子と同様に取り扱うことが可能です。しかし厳密に言えば、それらは等価ではありません。

具体的には、例えばクローンである複数の MXF ファイルが一つの MpUmid を共有できることは、UMID 応用原理の Principle 4 (UMID Identification) から明らかです。

また、MXF OP1c 形式など特殊な MXF ファイルにおいては、一つの MXF ファイルに、異なる外部出力部分を指定した複数の MXF MP を持たせることが可能ですから、この場合、MXF MP が記述した外部エッセンスフローの各々を大域的一意に特定した複数の MpUmid が存在することとなります。すなわち、一つの MXF ファイルが複数の異なる MpUmid によって大域的一意に特定される結果となります。

FpUmid の役割

では、MXF ファイルの内部エッセンスフローを記述した MXF FP に付与される FpUmid には、どのような応用が可能でしょうか？

元々 FpUmid は、ある MXF ファイル内において、MXF MP に内部エッセンスフローを供給する MXF FP を指定するための識別子に用いられてきました。従って FpUmid は、必ずしも大域的一意である必要はなく、当該 MXF ファイル内で一意であれば十分であることが判ります。

これより FpUmid は、2.3.2 節で紹介した関連付けツールとして活用することも可能です。図 18 は、この発想に基づき、派生素材の FpUmid を、その生成元となった素材を大域的一意に特定する MpUmid を継承することで生成した例を示したものです。

図 18 において、左部のオリジナル素材は、例えばカムコードを用いた撮影にて全く新たに得られた MXF ファイルであり、この場合その MpUmid 及び FpUmid には、共に新規生成された UMID 値(それぞれ新規生成された Mat.#値とゼロ Inst.#値で構成された“U₁”及び“U₂”)を設定することになります。

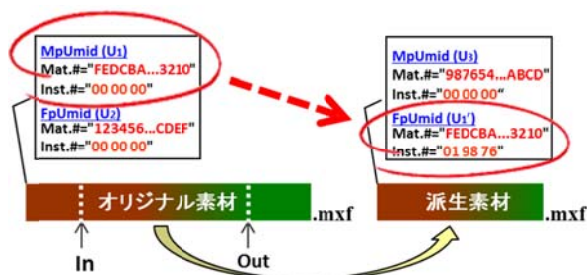


図 18 MXF における UMID の応用

いま、このオリジナル素材から所望の In/Out 点で指定される部分を切り出すことで、派生素材としての新たな MXF ファイルを生成したとします。この場合、当該派生素材もまた新たに生成された素材に他なりませんから、それ自体は独立した素材として取り扱えるよう、当該 MXF ファイルの MpUmid には新規生成された UMID 値である“U₃”を設定します。

一方、当該 MXF ファイルの FpUmid には、その生成元素材であるオリジナル素材を一意に特定する MpUmid (U₁)を継承した UMID 値である“U₁”を付与しています。

具体的には、その Mat.#には、オリジナル素材の MpUmid の Mat.# である“FE_h DC_h BA_h ... 32_h 10_h”なる値をそのまま継承し、その Inst.#には、新規生成した“01_h 98_h 76_h”なる値を設定しています。なお 2.3.2 節の議論に従い、後者においては、先頭の 1byte にはそれがオリジナル素材から直接派生したものであることを示す“01_h”を、またはそれに続く 2byte には、当該 MXF ファイル内において当該 FpUmid の一意性を担保した値である“98_h 76_h”を生成、付与しています。

図 18 から明らかのように、このようにして生成された FpUmid からは、その Inst.#部をゼロに置き換えることで、その生成元素材(図 18 の場合は左部のオリジナル素材に相当)を大域的一意に特定する MpUmid を簡単に得ることができます。従って、これを入力値とした UMID 解決を 5.4 節で紹介した UMID 解決プロトコルで実施することで、当該生成元素材へのアクセスが容易に得られるようになる訳です。

一般に映像制作の現場においては、完パケあるいは編集済み素材の品質管理や手直しに伴う部分変更などを目的として、その映像制作の元となったオリジナル素材への容易なアクセスの実現が渴望されています。

本節で紹介した FpUmid の応用は、現行の MXF 技術はそのままに、単に「UMID 応用原理」や「UMID 解決プロトコル」といった UMID 応用に関する基本技術をきちんと確立するだけで、MXF ファイルとしての素材を用いた映像制作環境における現場の様々なニーズに応える

ことが可能であることを示す好例であるといえましょう。

5.5.5. MXFにおけるUMID 応用調査報告

5.5.1 節で述べたように、本節で紹介した内容を含む「MXFにおけるUMID 応用の調査報告書」[38]は、本稿執筆時点では上位委員会でのレビューの段階にあります。

本節で紹介したMXFの内部動作モデルやMpUmid、FpUmid応用の基本的な考え方はMXF技術として普遍的なものです。本節で紹介できたのは、MXFにおけるUMID 応用のほんの触りの部分でしかありません。

実際、MXF仕様[13]自体は非常に複雑であり、また理論上、ここで紹介した単純な構造以外にも、様々な構造を持ったMXFファイルの生成することが可能です。

そこで「SMPTE UMID 応用プロジェクト」では、このように多岐にわたるMXFファイルに対して、その各々に含まれるUMIDがどのように運用可能かを詳細に調査し、更には5.6.4 節で紹介するように、現在話題となっているメディア・ストリーミングの収録や送出にMXF技術が関係した場合についても考察を加えました(これより、当該報告書の正式なタイトルは、“Study of UMID Applicatins in MXF and Streaming Media”¹⁷としました)。

「MXFにおけるUMID 応用の調査報告書」[38]は、これまでの調査報告書とは異なり、その内容をそのままSMPTE標準規格にするという類のものではありません。

むしろ、「SMPTE UMID 応用プロジェクト」の検討結果を広くワールドワイドな業界エキスパートに問い掛けてフィードバックを得ることで、最大限の自由度を与えつつも高度な相互運用性を保証した「MXFにおけるUMID 応用」を実現するために業界標準規格にすべき「コア」となる部分を見出していくことが、今後の重要なステップになると考えています。

これより「MXFにおけるUMID 応用の調査報告書」は、上位委員会のレビュー後、SMPTE Technology Committee Reports [46]として広く一般公開することを予定しています。つきましては、ご関心を持っていただけましたら、是非そちらも併せてご参照下さい。

なお、簡単のため本節では触れませんでした。MXFファイルに含まれる「UMID型メタデータ項目」にはもう一つ、BodyUmidと呼ばれるものがあります。これについては5.6.4 節で紹介します。

5.6. 拡張UMIDとその応用

5.6.1. はじめに

本節では、拡張UMIDとそのMXF及びメディア・ストリーミングにおける応用について紹介します。

5.6.2. 拡張UMIDとは？

図19(a)に、拡張UMIDのフォーマットを示します[1]。

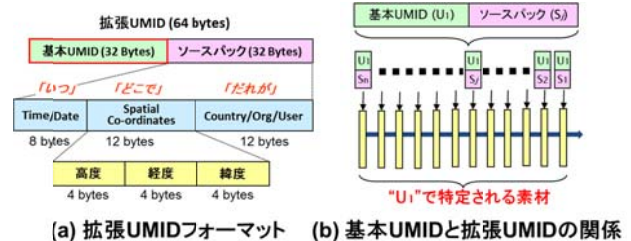


図19 拡張UMIDの構造と役割

拡張UMIDは、図1で紹介した32byteの基本UMIDに、ソースパックと呼ばれる32byteのバイト列を結合することで生成されます。

ここでソースパックとは、この拡張UMIDが付与された素材の初期生成に関する情報を記述しており、具体的には、当該素材の生成日時を記述した「いつ」(Time/Date)、当該素材の生成場所としての緯度、経度、高度情報を記述した「どこで」(Spatial Co-ordinates)、そして当該素材の生成主体情報を記述した「だれが」(Country/Org/User)から構成されています。

図19(b)に、基本UMIDと拡張UMIDとの関係を模式的に示します。3.3 節で紹介したように、VTRを用いた映像制作では、一回の記録開始終了(REC Start/Stop)で得られる連続したフレーム列を一つの素材とみなして一つのUMID(U_i)を付与し、またテープ上に記録された当該素材の任意位置へのアクセスにて当該UMIDが得られるよう、当該素材を構成する全てのフレームに対して当該UMID(U_i)を付与して記録することとしました。

一方、フレーム単位で変化する情報を記述したいとか、素材に含まれるある特定のフレームに対してのみメタデータを関連付けたいなどといった要求から、ある一つの素材に含まれる各々のフレームを区別する方法としてソースパックが導入されました。これより、ある素材の素材全体を大域的一意に特定する基本UMIDと、当該素材の内部において、フレームなど当該素材を構成する部分要素(これを「マテリアルユニット」と呼びます)を一意に特定するソースパックとを組み合わせた拡張UMIDを導入することで、例えば素材に含まれるあるフレームを大域的一意に特定することを可能にしました。

5.3.2 節で紹介したUMID 応用原理の Principle 6

¹⁷ その結果、当該報告書は、66ページに渡る大作になってしまいました。

(Extended UMID)は、図 19(b)に示したこの基本 UMID と拡張 UMID との階層化構造を規定したものです。

また同 Principle 7 (Source Pack)では、ソースパックは、あくまである(マテリアルユニットとしての)フレームを、「いつ」、「どこで」、「だれが」最初に生成したかを記述するために導入されたものがありますから、例えば映像制作に伴う素材編集において、オリジナル素材に含まれていたフレームを再利用した編集済み素材に新規生成 UMID を付与した結果、当該再利用フレームに付与された拡張 UMID の基本 UMID 部が当該新規生成 UMID 値に置き換えられたとしても、ソースパックは、原則としてオリジナル素材の生成時に最初に付与された値をそのまま継承することが求められています。

なお 5.6.4 節の図 21 では、この類似例として、カメラから拡張 UMID 付で生成、出力された一連のフレームで構成される入力ストリームを MXF ファイルとして収録する場合のソースパックの取り扱い方法について議論していますので、こちらと併せてご参照下さい。

ところで、拡張 UMID の特別な場合として、ソースパック全体をゼロ値で埋めたものや、そもそもソースパックを省略して先頭の基本 UMID 部のみで運用することも認められています。そこで本節では、拡張 UMID を、「ある素材を構成するマテリアルユニットの単位で付与された」UMID 一般を含む語句として用いることとします。

5.6.3. 拡張 UMID の応用 – 撮影場所情報の活用

5.6.2 節で紹介したように、素材が含む例えばフレームを大域的一意に特定することを目的に導入された拡張 UMID でしたが、3.3 節で紹介したように、特に素材識別子としての UMID 運用に自ずから限界があった VTR における UMID の応用においては、むしろソースパックのフレーム毎に変化する情報、特に移動時の撮影などでフレーム毎に変化する素材の生成場所に関する情報である「どこで」をコンパクトに記述する機能がより注目を集めることになりました。すなわち、ある素材に含まれる各々のフレームに拡張 UMID が付与されているということは、当該素材においてフレーム単位で変化する当該素材の撮影場所情報がフレームに同期して重畳されていることに他なりませんので、例えばそのような素材の再生において、それに同期した当該撮影場所情報の活用が簡単に実現できることとなります。

図 20 は、そのような拡張 UMID 応用の一例を示したものです。

図 20 では、台風の接近に伴って荒れる海岸の様子を

撮影した素材を、それに重畳された拡張 UMID と共に再生している様子を示しています。



図 20 拡張 UMID 「どこで」の応用

ここではフレームに同期して読み出された拡張 UMID の「いつ」に相当する撮影時刻情報を画面右上部のテロップにて、また同「どこで」に相当する撮影場所情報を、画面右部の地図内該当箇所への赤丸にて表示しています。

さて、図 20 はあくまで本節での事例紹介のために準備したのですが、実は拡張 UMID 自体は、2000 年に発行された SMPTE ST 330 の初版から導入されていた技術です。これより 3.3.1 節で紹介した HDCAM™ における UMID 機能の実装では、実際にソースパックも含めた拡張 UMID の取り扱いまでをサポートしました。

具体的には、2002 年に HDCAM™ カムコーダの専用アクセサリとしての GPS ユニット[47]を販売し、これを装着した HDCAM™ カムコーダで撮影した場合に、フレーム毎に、その撮影時刻と、その撮影場所としての緯度、経度、高度情報とを併せてテープに重畳記録できるようにしました。

ただ、図 20 で紹介したような応用を実現するに当たり大きな障壁になったのが、いわゆる地図データの手配でした。当時はまだ Google マップなどもなく、地図の専門業者からそれを取り寄せるのに高額のライセンス料が必要でした。他方で、素材は物理的なテープでしか存在しませんので、このような拡張 UMID が埋め込まれた素材を準備したところで、位置情報を活用したオンラインでの素材検索など実現する術もなく、結局はコストに見合ったメリットが得られないということでお蔵入りになってしまいました。

あれから 10 年以上が経ち、民生用では GPS 情報を地図データと併せて活用することは、ごく当たり前の手段となりました。他方で業務用については、筆者が知る限りにおいて、残念ながら未だそのような取り組みは殆ど見当たりません。

一方、特に大規模災害などの報道においては短時間に膨大な数の素材が生成されますので、その仕分けを人手

に頼っているのは多大なコストが掛かってしまうことは、東日本大震災の報道体制において日本の全ての放送局が得た大きな教訓であったかと思えます。

今や素材もファイル化され、ネットワーク上で自由に検索できる環境が整いつつあります。この状況において、人手を介さず自動的に付与することが可能であり、素材の基本中の基本情報である「いつ」、「どこで」、「だれが」当該素材を生成したかを記述した拡張 UMID は、先述したような短時間に大量の素材が生成されるような状況においてそれらの素材を効率的に取り扱うにあたり、少なくとも最初の自動仕分けの段階で必要なメタデータ技術の最有力候補であると考えています。

今後の映像制作の現場における拡張 UMID の活用に、大いに期待したいと思えます。

5.6.4. 拡張 UMID の応用 – MXF ファイル収録/再生

5.5.3 節では、MXF ファイルに含まれる「UMID 型メタデータ項目」として、MpUmid と FpUmid を紹介しました。これらは 2.2 節で紹介した 32byte の基本 UMID フォーマットに則ったもので、いずれも MXF ファイルのヘッダ部分に格納されています。

一方、5.5.5 節で触れたように、MXF ファイルに含まれる「UMID 型メタデータ項目」にはもう一つ、BodyUmid と呼ばれるものがあります。これは、MXF のファイルのボディ部分において、5.5.2 節で紹介した「汎用コンテナ」によって、フレームなどエッセンスデータの区分けの単位であるエジットユニット毎に付与されるかたちで格納されています。

そして BodyUmid は、一般的には拡張 UMID の形式にて、それが付与された(エジットユニットとしての)フレームを、「いつ」、「どこで」、「だれが」最初に生成したかについて記述しています。

さて、5.6.2 節では拡張 UMID が付与される単位としての「マテリアルユニット」を、また本節では(拡張 UMID 形式の)BodyUmid が付与される単位としての「エジットユニット」を紹介しました。ただ、多くの場合、これらはいずれもフレームに相当します。

これより、入力ストリームを収録し新たな MXF ファイルを生成する場合、当該入力ストリームを構成する(マテリアルユニットとしての)フレームに拡張 UMID が付与されていたならば、当該フレームを(エジットユニットとして)保存する際に、当該拡張 UMID を BodyUmid として一緒に保存することが期待されます。

図 21 は、その一例として、カメラが撮影し SDI 出力

した映像を MXF レコーダで受けて、新たな MXF ファイルを生成する様子を示したものです。

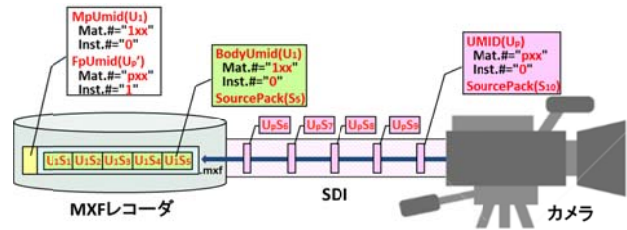


図 21 拡張 UMID 付入力からの MXF ファイル生成

図 21 において、カメラは、GPS ユニットなどが装着されていることを前提に、それが生成する出力ストリームを構成する各々のフレームに対して、その生成時刻及び生成場所を記述した拡張 UMID を付与した上で出力しています。ここで当該拡張 UMID は、「pxx」なる Mat.# とゼロ値の Inst.# から構成される「U_p」なる基本 UMID 部¹⁸ と、フレーム毎に少なくとも生成時刻が異なった「S_j」なるソースパックから構成されています。

これを受信した MXF レコーダは MXF ファイルを生成しますが、この時当該 MXF ファイルの MpUmid には、自らを大域的一意に特定できるよう、新規生成された「1xx」なる Mat.# とゼロ値の Inst.# から構成される「U₁」なる基本 UMID 値が設定されます。他方で同 FpUmid には、入力ストリームに付与された拡張 UMID の Mat.# から継承された「pxx」なる Mat.# 値と、新たに新規生成した「1」なる Inst.# 値から構成される「U_p」なる基本 UMID 値が設定されています。

そして BodyUmid ですが、オリジナル素材として新規生成された MXF ファイルの場合、UMID 応用原理の Principle 6 (Extended UMID) により、その基本 UMID 部は原則として当該 MXF ファイルを大域的一意に特定すべく付与された MpUmid 値と同じ値(ここでは「U₁」)にします。他方ソースパックについては、同 Principle 7 (Source Pack) に従い、保存対象の入力フレームに付与されていた拡張 UMID のそれをそのまま継承します。

この結果、例えば「U_pS₁₀」なる拡張 UMID が付与された(マテリアルユニットとしての)フレームを受信し、これを(エジットリストとして)記録した場合、当該フレームに対して「U₁S₁₀」なる BodyUmid を生成、付与して、共に記録することになります。

一方、既存の MXF ファイルを再生して出力ストリームを生成する場合、当該 MXF ファイルに含まれる(エジットリストとしての)フレームに BodyUmid が付与され

¹⁸ このようなライブストリームのフレームに付与される基本 UMID の生成方法の仕様には曖昧な部分が残っており、現在その明確化を進めています。

ていたならば、当該フレームを(マテリアルユニットとして)再生出力する際に同 BodyUmid を拡張 UMID として付与して一緒に出力することが期待されます。

図 22 は、その一例として、図 21 で得られた MXF ファイルを MXF プレーヤで再生し、SDI 出力する様子を示したものです。

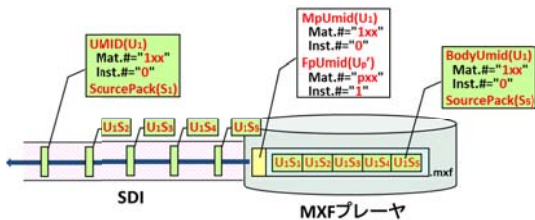


図 22 BodyUmid 付 MXF ファイルの再生

図 22 において、MXF プレーヤは、当該 MXF ファイルの再生の際にそれが生成する出力ストリームを構成する各々のフレームに対して、当該 MXF ファイル内の対応するフレームと一緒に記録されていた BodyUmid を読み出し、それをそのまま拡張 UMID として当該各々のフレームに付与した上で出力しています。この結果、出力ストリームに含まれるフレームに付与された拡張 UMID の基本 UMID 部には、そのストリーム生成の元となった MXF ファイルを大域的一意に特定する MpUmid の値が入った状態で出力されることとなります。

これより、そのような出力ストリームの受信した場合、そのフレームに付与されている拡張 UMID の基本 UMID 部を取り出し、5.4.2 節で紹介した UMID 解決プロトコルを用いて当該 UMID を解決することで、受信した出力ストリームを生成する元となった MXF ファイルへのアクセスが実現できることとなります。

5.6.5. 拡張 UMID の応用 – ストリームの品質管理

拡張 UMID の最後の応用事例として、拡張 UMID が付与されたマテリアルユニットとしてのフレーム列で構成されるライブストリームにおいて、2.3.2 節で紹介した Inst.# における世代番号を運用する事例を紹介します。

2.3.2 節によれば、オリジナル素材からの派生素材の生成の際に用いられる関連付けツールとしての UMID には、オリジナル素材からの世代番号をその Inst.# に設定する機能が選択肢としてありました。

元々これは、3.3.2 節で紹介したように、VTR のダビングに伴う画質劣化を管理すべく、オリジナル素材からのダビング回数を世代番号として Inst.# に記録するというものでした。VTR においては、UMID は全てのフレームに対して付与され、フレーム単位で伝送、記録されて

いましたので、これは実質的には、素材を構成する各々のフレームに対して、それがオリジナル素材に含まれていた時から施された圧縮伸張処理の回数を Inst.# に保持していたことに他なりません。

そこで、その適用を単に記録時の圧縮伸張処理のみならず、伝送中に施される画像処理も含めた任意の画像処理に拡張することで、例えばライブストリームにおける映像の品質管理が可能となります。

図 23 は、その一例として、二台のカメラからの SDI 出力を切り替えて一本のライブストリームを生成する様子を示したものです。

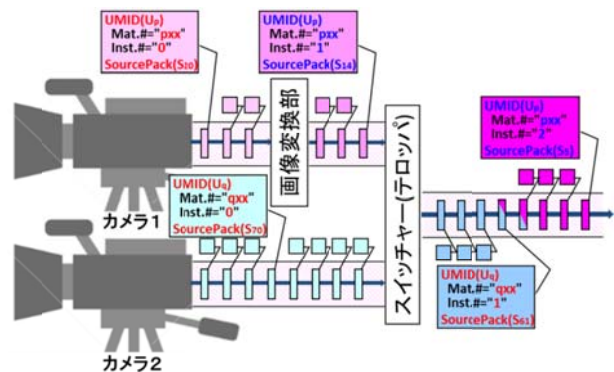


図 23 ライブストリームにおける世代番号の運用

図 23 において、各カメラは、図 21 と同様それぞれが生成する出力ストリームを構成する各々のフレームに対して、拡張 UMID を付与した上で出力しています。

ここでカメラ 1 及びカメラ 2 から出力されたフレームには、それぞれ“pxx”、“qxx”なる Mat.# と、共にゼロ値の Inst.# から構成される“U_p”、“U_q”なる基本 UMID 部と、“S_j”なるソースパックとで構成される拡張 UMID が生成、付与されています。

また図 23 では、カメラ 2 からの出力ストリームが「スイッチャー(テロップ)」へ直接入力されているのに対し、カメラ 1 からの出力ストリームは一旦「画像変換部」へ入力され、SD から HD への画角変換や色調整など何らかの画像変換処理が施された上で「スイッチャー(テロップ)」へと入力されています。

そして「スイッチャー(テロップ)」は、それらの入力ストリームを、二フレーム分の遷移領域を介して切り替えた他、全てのフレームに対してロゴなど何らかのテロップ表示処理をおこなって出力しています。

このようなライブシステムにおいて、フレームの画像処理を実施する画像変換部及びスイッチャーが、自らの画像処理の足跡を残すべく各フレームに付与されている拡張 UMID の Inst.# に含まれる世代番号をインクリメント(1 増加)してやれば、3.3.2 節で紹介した VTR での映像

制作における素材の品質管理と同じようなことが、ライブストリームの映像でも実現可能となります。

例えば図 23 の場合、「画像変換部」、「スイッチャー(テロップ)」共に、入力フレームに付与されていた拡張 UMID を、その Inst.#の世代番号をインクリメントさせたものに置き換えて出力しています。その結果、スイッチャーからは、カメラ 1 からのフレームには世代番号“2”が、またカメラ 2 からのフレームには世代番号“1”が含まれた拡張 UMID がそれぞれ付与されて出力されています。

なお、切り替えの遷移領域においては、一つの出力フレームが、カメラ 1、カメラ 2 の両方からの入力フレームを合成することで生成されます。このような場合、基本的には当該出力フレームの合成においてデータ量的に寄与が大きい入力フレームの拡張 UMID を継承することになります。すなわち図 23 の場合、遷移領域の先頭(右端)の出力フレームはカメラ 1 からの入力フレームを、同末尾(左端)の出力フレームはカメラ 2 からの入力フレームをベースとして出力フレームが合成されたとみなし、その結果、それぞれ“pxx”、“qxx”なる Mat.#を継承した拡張 UMID が付与されることとなります。

また図 23 の事例では、スイッチャーが全フレームに対してテロップ表示処理をおこなっていることを前提に、全てのフレームに付与された拡張 UMID の世代番号をインクリメントさせています。しかし、もしスイッチャーが単に入力フレームを何ら加工することなく出力フレームとして出力した場合は、それに付与された拡張 UMID もまた、そのまま変更なく出力されることとなります。ただし VTR のように、フレーム伝送自体が非可逆的な圧縮伸張処理を伴っている場合は、それ自体を拡張 UMID の世代番号管理に反映させることが必要です。

ライブストリームを生成、処理する全ての映像機器がこのような拡張 UMID の取り扱い機能をサポートすれば、それらで構成されるライブシステムでは、システム内の任意の場所においてフレームに付与された拡張 UMID の Inst.#値を検出することで、ライブストリームの映像品質のフレーム単位でのモニタリングが可能となります。

また本事例で明らかのように、カメラが最初に生成、付与した Mat.#値は、当該ライブシステムの後段でも常に継承されます。そして、Mat.#値には MAC アドレスなど当該カメラを大域的一意に特定することが可能な情報が含まれているのが一般的です。従って、フレームに付与された拡張 UMID の Mat.#を検出することで、当該フレームがどのカメラによって撮影されたかなど、素材を最初に生成した映像機器をフレーム毎に特定することも

可能となります。

6. まとめに替えて

本稿では、放送・映像メディア業界の業界標準映像素材識別子である UMID 応用の実用化と更なる展開を実現するために筆者らが SMPTE を舞台に 2011 年から取り組んでいる「SMPTE UMID 応用プロジェクト」について、本稿執筆時点(2015 年 8 月)での全容を紹介しました。

まずは 2014 年度の映像情報メディア未来賞/フロンティア賞そして本稿執筆の貴重な機会を与えて下さいました映像情報メディア学会に、紙面をお借りして心よりお礼を申し上げます。また、本稿を最後までお読み下さいました読者の方々にも厚く感謝いたします。

さて、近年、業界で最もホットな話題といえば、4K/8K に代表される UHD (Ultra High Definition) と、SDI に替わる放送局内の映像伝送インフラを狙った VoIP (Video over Internet Protocol) であると思われます。そして、特に前者については、我が国がワールドワイドな業界の主導的立場にあることは、よく知られているところです。

一方、後者を見れば、既に我が国の幾つかの業界ベンダがそのサポートを謳ってはいるものの、例えば JT-NM (Joint Task Force on Networked Media) [48] など、有力ユーザを含む業界プレーヤ全体を巻き込んだワールドワイドな「大きな流れ」といったところでは、残念ながら我が国のプレゼンスは殆どないと言わざると得ません。

これまで放送・映像メディア業界は、例えば 2000 年末に始まった HD 放送の素材伝送に必要な 1.5Gbps の伝送容量を確保するために HD-SDI [49] を開発するなど、業界が自らが必要とする技術を自らで開発、実用化するという選択肢しか持ちませんでした。しかし今や、HD-SDI の 5 倍以上の伝送容量を持つ 10Gb イーサーネットが簡単に手に入る時代となりました。

もちろん、フレーム単位での切り替えといった業界特有の要求仕様については専用技術に一日の長はありますが、伝送容量といった基本仕様に既にそれだけの差があること、また ICT 機器と映像関連機器ではその市場規模に 100 倍以上の差があり [50]、それに伴う R&D 投資額の違いや大量生産に伴うコストダウン効果などを踏まえると、今後は基本仕様の部分は ICT 市販品いわゆる COTS (Commercial Off-The-Shelf) の利用を前提に、上位アプリに近い部分にて業界特有の課題解決を図るといったアプローチが最も妥当であり、前出の SDI に替わる VoIP の確立とはまさにこの流れに沿ったものといえます。

一方、その文脈で言うならば、3.2 節で紹介したように、

ICT を駆使した効率的なファイルベース映像制作の実現のためにファイル書式なる部分で業界特有の MXF を開発、導入したことはまさに慧眼であり、現在、業界の素材交換において、MXF が事実上の標準化の地位を確固たるものにしたことは当然の帰結であったといえましょう。

そして次に期待されるのは、ネットワーク環境においてこのような素材を効率的に取り扱うための仕掛けであり、第4章で議論したように、UMID はまさにそのインフラ技術として最右翼にあるといえます。

本稿で紹介したように、「SMPTE UMID 応用プロジェクト」は、UMID 応用の実用化を実現する上で解決すべき課題の明確化と、当該課題の一つであった「UMID 応用原理」なる最も基本的な部分の検討を終え、目下、「UMID 解決プロトコル」や「MXF における UMID 応用」といった、より実践的な部分での検討に取り組んでいるところです。

第1章そして本章の冒頭でも述べましたが、VTR を中心としたベースバンドの時代にはワールドワイドな映像技術を牽引してきた我が国ではありましたが、業界の先端技術動向がファイルベースや VoIP といった ICT の応用分野に向かうにつれ、カメラなど一部の例外を除いて我が国のワールドワイドな業界プレゼンスは大幅に低下し、誤解を恐れず言えば、業界における ICT を活用した映像制作分野でのほぼ全ての技術の仕様決めなどは、事実上、欧米勢に独占されてきました。

本稿で紹介した筆者らの取り組みは、微力ながらこのような状況に一石を投じる効果はあるものと考えています。つきましては、もし本活動にご興味を持っていただけたら、是非、筆者までご連絡をお願いします¹⁹。

¹⁹ 本稿に関する質問、コメントなどがございましたら、info@metafrontier.jp まで、ご連絡下さい。

Reference

- [1] SMPTE ST 330
- [2] SMPTE RP 205:2014
- [3] <https://www.smppte.org/>
- [4] 2012 年 ITE 年次大会予稿集 10-5
- [5] 2013 年 ITE 年次大会予稿集 14-3
- [6] 2014 年 ITE 年次大会予稿集 1-6
- [7] SMPTE RP 210
- [8] SMPTE ST 336
- [9] ISO/IEC 11578
- [10] ISO 15706
- [11] <http://eidr.org/>
- [12] <https://tech.ebu.ch/docs/techreview/ebu-smppte-tf-bitstream.pdf>
- [13] SMPTE ST 377、他
- [14] <http://www.sony.jp/hdcam/>
- [15] SMPTE ST 291
- [16] <http://www.sony.jp/xdcam/>
- [17] <http://www.w3.org/TR/xml/>
- [18] ISO/IEC 15938-1
- [19] Y.Shibata, et al., "Introduction to XDCAM Metadata", Proc. NAB2005 BEC、pp.421-424 (2005)
- [20] <http://fims.tv/>
- [21] 柴田賀昭、八木伸行：“FIMS”、映情学誌、66、10、pp.835-841 (2012)
- [22] https://kws.smppte.org/kws/projects/project/details?project_id=90
- [23] Y.Shibata and J.Wilkinson, "UMID Applications in Practices", Proc. SMPTE 2011 ATC, also available at: <http://metafrontier.jp/drupal/sites/default/files/papers/umidApp4SmppteTechConf111025.pdf>
- [24] <https://www.youtube.com/watch?v=Us6eVtGq7mA>
- [25] <http://www.metafrontier.jp/drupal/sites/default/files/smppteHome111125.png>
- [26] Y.Shibata and J.Wilkinson, "UMID Applications in Practices", SMPTE Mot. Imag. J.; Mar. 2012; 121:(2), pp.58-67
- [27] <http://metafrontier.jp/drupal/sites/default/files/info/umidApp4NewsReleaseJ120411.pdf>
- [28] M. De Geyter, et al., "Integration Demands on MAM Systems: A Proof of Concept Solution", SMPTE Mot. Imag. J.; Nov./Dec. 2008; 117:(8) 38-46
- [29] <http://metafrontier.jp/drupal/sites/default/files/info/agendaAndIntr2Umid120913.pdf>
- [30] <http://metafrontier.jp/drupal/sites/default/files/papers/umid4Fims120418J.pdf>
- [31] Y.Shibata and J.Wilkinson, "Latest Status of UMID and its Applications in File-based Workflow", BroadcastAsia 2012 International Conference (June 2012), available at: <http://metafrontier.jp/drupal/sites/default/files/papers/umidApp4Bca120619Ver3.pdf>
- [32] <http://metafrontier.jp/drupal/sites/default/files/papers/umidApp4Mam4AbuTc121013.pdf>
- [33] Y.Shibata, "Latest Status of UMID Application Project in SMPTE", Proc. SMPTE 2013 ATC, also available at: <http://metafrontier.jp/drupal/sites/default/files/papers/umidApp4SmppteTechConf131023.pdf>
- [34] <http://metafrontier.jp/drupal/sites/default/files/info/summaryOfStudyReportOnUMIDApplicationsPart1.pdf>
- [35] https://kws.smppte.org/kws/projects/project/details?project_id=174
- [36] <http://metafrontier.jp/drupal/sites/default/files/info/summaryOfStudyReportOnUMIDApplicationsPart2-1.pdf>
- [37] https://kws.smppte.org/kws/projects/project/details?project_id=273
- [38] <http://metafrontier.jp/drupal/sites/default/files/info/summaryOfStudyReportOnUMIDApplicationsInMxf.pdf>
- [39] <http://metafrontier.jp/drupal/sites/default/files/info/summaryOfStudyReportOnUMIDApplicationsPart2-2.pdf>
- [40] IETF RFC 1034、他
- [41] IETF RFC 6762
- [42] IETF RFC 6763
- [43] IETF RFC 3650
- [44] N.Wells, et al., "The MXF Book", Focal Press (ISBN-13: 978-0240806938)
- [45] SMPTE ST 379
- [46] SMPTE TC-30MR STUDY GROUP REPORT – "Study of UMID Applications in MXF and Streaming Media", to be published at: <https://www.smppte.org/standards/reports>
- [47] <http://www.sony.jp/products/Professional/ProMedia/arc/020701a.html>
- [48] <http://www.jt-nm.org/>
- [49] SMPTE ST 292
- [50] A.Kovalick, "The State of the Media Facility", SMPTE Mot. Imag. J.; Sep. 2013; 122:(6), pp.86-91