SMPTE RECOMMENDED PRACTICE

Application of Unique Material Identifiers in Production and Broadcast Environments

Table of Contents

Page

Approved
August 15, 2014

Foreword

SMPTE (the Society of Motion Picture and Television Engineers) is an internationally-recognized standards developing organization. Headquartered and incorporated in the United States of America, SMPTE has members in over 80 countries on six continents. SMPTE's Engineering Documents, including Standards, Recommended Practices, and Engineering Guidelines, are prepared by SMPTE's Technology Committees. Participation in these Committees is open to all with a bona fide interest in their work. SMPTE cooperates closely with other standards-developing organizations, including ISO, IEC and ITU.

SMPTE Engineering Documents are drafted in accordance with the rules given in its Standards Operations Manual.

SMPTE RP 205 was prepared by Technology Committee 30MR on Metadata and Registers.

Intellectual Property

SMPTE draws attention to the fact that it is claimed that compliance with this Standard may involve the use of one or more patents or other intellectual property rights (collectively, "IPR"). The Society takes no position concerning the evidence, validity, or scope of this IPR.

Each holder of claimed IPR has assured the Society that it is willing to License all IPR it owns, and any third party IPR it has the right to sublicense, that is essential to the implementation of this Standard to those (Members and non-Members alike) desiring to implement this Standard under reasonable terms and conditions, demonstrably free of discrimination. Each holder of claimed IPR has filed a statement to such effect with SMPTE. Information may be obtained from the Director, Standards & Engineering at SMPTE Headquarters.

Attention is also drawn to the possibility that elements of this Standard may be subject to IPR other than those identified above. The Society shall not be responsible for identifying any or all such IPR.

Introduction

This section is entirely informative and does not form an integral part of this Engineering Document.

The Unique Material Identifier (UMID) is a unique audiovisual material identifier standardized by SMPTE as SMPTE ST 330.

In 1998, the EBU/SMPTE Task Force for Harmonized Standards for the Exchange of Programme Material as Bitstreams recommended that a unique material identifier and a single, generic, file wrapper were needed to facilitate the exchange of program material and metadata as bit-streams. These recommendations were realized with the standardization of the Unique Material Identifier (UMID) in the year 2000 and later, with the standardization of the Material eXchange Format (MXF) in the year 2004.

While the UMID itself is independent of any material format or packaging, it was a natural consequence that the UMID was adopted as the mandatory component in the MXF files by which the files, and the material they contain, are uniquely identified. Thus, as MXF has gained an ever increasing role within file-based workflows, the UMID has also become more widely accepted within the industry.

However, in spite of its wide adoption, the usefulness of the UMID as a globally unique material identifier has been very limited. There have been experimental trials and media products that have utilized the UMID as the material identifier by which external metadata is uniquely associated with material. Despite this original intent upon its introduction, the UMID could not effectively be used as a common means to uniquely identify material across applications and vendor products. This failing can be attributed to the following reasons:

Problem 1: No industry wide trust framework has been established, by which the uniqueness and trustworthiness of a UMID can be verified, ,

Problem 2: No industry wide criteria has been defined by which an application can use to determine when a UMID can remain unchanged and when a new UMID must be assigned,

Problem 3: No industry common method has been defined by which a UMID could be resolved to its descriptive and location information, describing and providing access to the identified material.

This Recommended Practice is provided to address the Problems 1 and 2 above.

The primary intent of the UMID is to be a globally unique material identifier. However, the global uniqueness of a UMID can be easily violated if proper, proactive, treatment of the UMID is not practiced. For example, suppose a material asset is copied from one location to another (a.k.a. cloned) and some portion of the picture essence is overwritten, removed, or changed but the UMID is not updated. The resulting material asset will differ from the original, but will be improperly assigned the same UMID identity. To prevent such a scenario from occurring, a UMID must be reassigned in accordance with a common set of principles.

If one creates a new UMID based on SMPTE ST 330 specification, the UMID's global uniqueness is algorithmically guaranteed. Therefore, the integrity of the UMID as a globally unique material identifier is guaranteed, within practical limits, only when one always assigns a newly created UMID to a material asset. This is however equivalent with the use of a proprietary material identifier, which is defined within the scope of a closed application domain, and does not provide the benefits of using a UMID at all.

So what are the benefits of using a UMID as an industry standard unique material identifier?

One of the important benefits is that material assets can be managed, seamlessly, over multiple media products, provided those media products utilize the UMID and adhere to the same UMID creation and reuse rule set. If this is realized, and if more than one material asset is identified by the same UMID, these material assets can be trusted to be identical, regardless of the product managing those material assets. Furthermore, this scenario can be expanded to cover an entire media production system; using the UMID as a common unique material identifier across a heterogeneous system, comprised of numerous media products from different vendors. However, in order to realize this, a universal rule set for UMID applications that provides a means by which the integrity of a UMID attached to a given material asset can be trusted, regardless of the products used to create the UMID and/or attach the UMID to the material asset, is required.

Unfortunately, SMPTE ST 330 does not specify nor provide the set of rules required to facilitate the trust framework to guarantee the integrity of a UMID. Therefore additional regulations and rules must be defined to create this trust framework.

In this Recommended Practice, the UMID Application Principles are specified as a set of rules applicable to all UMID applications, which establishes a trust framework for UMID usage. If a media product strictly adheres to the UMID Application Principles, the integrity of the UMIDs attached to material obtained from that product can be trusted, such applications and products are known as “UMID-aware”.

Since the means by which the UMID Application Principles are applied depend on the type of product and the type of recording medium used, the Principles are specified in terms of “What to Achieve” versus “How to Achieve Them”, and how they are achieved is left up to the implementer of the product. Please refer to Annex B for a typical implementation example of the UMID Application Principles using a UMID as a globally unique material identifier.

Furthermore, since SMPTE ST 330 allows UMIDs to be used for purposes other than globally unique material identity, rules for these additional purposes have been defined within the UMID Application Principles.

In summary, the purpose of this Recommended Practice is to establish a common trust framework by which applications can utilize UMIDs in a consistent, unambiguous, and well-understood manner that allows for the maintenance of the logical relationships between material assets, their metadata, and their instances.

1 Scope

This Recommended Practice defines the UMID Application Principles that must constitute the basis of any UMID application, and recommends several points on UMID use that must be carefully considered for each application. It also provides informative text on the UMID basics, how to implement the UMID Application Principles for a UMID as a globally unique material identifier, various UMID application examples, and frequently asked questions on the UMID applications.

2 Conformance Notation

Normative text is text that describes elements of the design that are indispensable or contains the conformance language keywords: "shall", "should", or "may". Informative text is text that is potentially helpful to the user, but not indispensable, and can be removed, changed, or added editorially without affecting interoperability. Informative text does not contain any conformance keywords.

All text in this document is, by default, normative, except: the Introduction, any section explicitly labeled as "Informative" or individual paragraphs that start with "Note:”

The keywords "shall" and "shall not" indicate requirements strictly to be followed in order to conform to the document and from which no deviation is permitted.

The keywords, "should" and "should not" indicate that, among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.

The keywords "may" and "need not" indicate courses of action permissible within the limits of the document.

The keyword “reserved” indicates a provision that is not defined at this time, shall not be used, and may be defined in the future. The keyword “forbidden” indicates “reserved” and in addition indicates that the provision will never be defined in the future.

A conformant implementation according to this document is one that includes all mandatory provisions ("shall") and, if implemented, all recommended provisions ("should") as described. A conformant implementation need not implement optional provisions ("may") and need not implement them as described.

Unless otherwise specified, the order of precedence of the types of normative information in this document shall be as follows: Normative prose shall be the authoritative definition; Tables shall be next; followed by formal languages; then figures; and then any other language forms.

3 Normative References

Note: All references in this document to other SMPTE documents use the current numbering style (e.g. SMPTE ST 298:2009) although, during a transitional phase, the document as published (printed or PDF) may bear an older designation (such as SMPTE 298-2009). Documents with the same root number (e.g. 298) and publication year (e.g. 2009) are functionally identical.

The following standards contain provisions which, through reference in this text, constitute provisions of this recommended practice. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this recommended practice are encouraged to investigate the possibility of applying the most recent edition of the standards indicated below.

SMPTE ST 298:2009, Universal Labels for Unique Identification of Digital Data

SMPTE ST 330:2011, Unique Material Identifier (UMID)

4 UMID Application Principles

4.1 Introduction

This section defines a set of UMID Application Principles. Except for this and Section 4.2, each subsection starts with the statement of the principle, followed by implications that will help the reader to obtain a better understanding of the statement.

4.2 Principle 1 – Definitions

For the purpose of describing the UMID Application Principles, the capitalized terms used in the statements below shall have the respective meanings defined as follows:

- **Basic UMID** shall mean the UMID of 32-byte long composed of its12-byte UL, the 1-byte length field, 3-byte Instance Number Field and 16-byte Material Number Field, specified in SMPTE ST 330. While this term may represent either on its own or the first 32-byte of Extended UMID, the phrase "the Basic part of Extended UMID" may be used to refer to the latter case for its further clarification if needed.

- **Extended UMID** shall mean the UMID of 64-byte long composed of its 32-byte Basic UMID followed by 32-byte Source Pack, used to identify a finer granularity than is identified by its Basic part, specified in SMPTE ST 330.

- **Instance** shall mean a specific item of stored material(s) that shares the same Material Number regardless of its Instance Number value.

- **Instance Number** shall mean the value in the Instance Number Field of the UMID. The Instance Number shall take a value of zero for the original Material and a non-zero value for any copy or derivation of the original Material.

  Note: SMPTE ST 330 defines several methods by which the Instance Number value is created. The uniqueness of an Instance Number value is guaranteed only within a predefined closed domain where the uniqueness of an Instance Number is appropriately controlled by using such as local registration. In open domains, the assignment of a new non-zero Instance Number value cannot guarantee a unique value; *i.e.*, there is a small, but finite, probability that duplicate values can occur.

- **Instance Number Field** shall mean the 24 bits field using bytes 14^{th}-16^{th} of the UMID as specified in SMPTE ST 330.

- **Instance Number Generation Method Field** shall mean the field of the 4 least significant bits (LSBs) of byte 12^{th} of the UMID UL used to identify the method to create the Instance Number field value as specified in SMPTE ST 330. Note that the special value of 15 (F_h) is reserved for this field in order for the UMID with it to be used to signal a live-stream source that has never been fully recorded (and hence has never been instantiated as a persistent and deterministic form of material) rather than to be used as a unique material identifier.

- **Material** shall mean a persistent and deterministic form of unique set of audio and/or visual essence playable on a single timeline, which may be a single item or a set of synchronized items, which shall be continuous along the timeline. Material, often preceded with the term "original", shall be globally uniquely identified by a UMID with a newly created Material Number together with zero Instance Number, whose global uniqueness is algorithmically guaranteed according to SMPTE ST 330. The essence in the Material may either be a playable bit stream, or an abstract source from which a playable bit stream may be created.

- **Material Number** shall mean a non-zero value for the Material Number Field in the UMID.

- **Material Number Field** shall mean the field of bytes 17^{th}-32^{nd} of UMID as specified in SMPTE ST 330.

- **Material Unit** shall mean the quantum of material composed as its cyclic sampling structure.

- **Source Pack** shall mean the last 32 bytes of the Extended UMID, which contains the information of "when", "where" and "who" has created the material, as specified in SMPTE ST 330.

- **UL** shall mean the SMPTE Universal Label specified in SMPTE ST 298, which is truncated to 12 bytes length when used in the UMID format. When the UMID UL is extracted for external use, it shall be padded to 16 bytes by appending 4 bytes of zero value and shall change the value of the label size in byte 2nd from $0A_h$ to $0E_h$.

- **UMID** shall mean the Unique Material IDentifier whose format and value creation methods are specified in SMPTE ST 330. The UMID may take the form of either 32-byte Basic UMID or 64-byte Extended UMID, where the Extended UMID is still a UMID, but the one extended with additional 32 bytes of geo-location and other information to form the Source Pack.

- **UMID Managed Domain** shall be an authoritative source in which the uniqueness and meaning of a UMIDs Material Number and Instance Number shall be managed and guaranteed. A UMID Managed Domain may range from a single registry, within a single location, to a large federated registry spanning multiple organizations. UMID Managed Domains shall govern the Law of Identity for UMIDs within their scope of authority.

4.3 Principle 2 – UMID Creation

A UMID based material management system shall create a UMID with a newly created Material Number together with a zero Instance Number in accordance with SMPTE ST 330 for original Material at its point of entry into the UMID Managed Domain.

Implications

- Original Material may be an essence bit stream of its highest quality as a master, or an abstract source from which actual playable bit stream is to be created.

- Material may take the form of a bounded sequence of frames, a single frame or a still image, a bounded sequence of interleaved audiovisual essences, or audio signal occurring on a single timeline.

- A newly created UMID with a zero Instance Number value, based on SMPTE ST 330, is algorithmically guaranteed to be globally unique within practical limits, and therefore only Material with this UMID may be consistently managed in a global sense.

- Source Pack containing "when" "where" and "who" information at the entry may be appended as needed to any Basic UMID with the proviso that the Source Pack data values represent the best effort to establish those values as true and meaningful.

4.4 Principle 3 – UMID Integrity

Different original Materials shall be globally uniquely identified by different UMIDs.

Implications

- The strict definition of "different" shall be understood by the antonym of "identical" used in the Principle 4 (UMID Identification),

- In general, a UMID created somewhere unknown may not be guaranteed to be globally unique even with its zero Instance Number because of a possibility that material that breaches the UMID Integrity might exist elsewhere unknown,

- A UMID with a non-zero Instance Number may not be guaranteed to be unique for all Instances of the same Material except in certain closed and controlled environments. This is because it is finitely possible to create a new Instance of Material with the same Instance Number value using the same number generation method. The specific case where each Instance Number value is guaranteed to be unique for a given Material Number value is where the Instance Number values are assigned based on a common registry. In the general case, a new entry is required for each Instance to be consistently managed as a new Material, resulting in the UMID being replaced with a newly created value of a new Material Number and a zero Instance Number.

- A material with a UMID that signals the “Live stream” value (F_h) at the Instance Number Generation Method Field may not be uniquely determined because it is open-ended (a.k.a. the file is ‘growing’). A new entry shall be required for the resulting persistent and deterministic material to be consistently managed in a global sense, resulting in the UMID being replaced with a newly created value of a new Material Number and a zero Instance Number.
- If a device creates more than one item of recorded material simultaneously, it shall be the responsibility of the device to assign different UMIDs, each with either a new different Material Number value and zero Instance Number or the same Material Number but different Instance Number, in order to unambiguously distinguish them, unless they are proven to be identical.

4.5 Principle 4 – UMID Identification

If more than one material is uniquely identified by a single UMID, their representations at playout shall be identical bit by bit on the timeline.

<u>Implications</u>

- The UMID shall be understood to uniquely identify a material representation at its playout, which is intrinsic to the material regardless of its specific instantiation stored in a persistent and deterministic fashion.
- Identical materials may share the same UMID value (as it occurs in a simple copy operation).
- The materials in different mathematically lossless compression schemes may share the same UMID if their playout representations are identical.
- Materials of different bit streams resulted from such as padding/filler items insertion, different descriptive metadata or file wrapper of the Materials, may share the same UMID if their playout representations are identical.
- Playable data such as the edit decision list (EDL) or the playlist may share the same UMID as that to be attached to the Material generated from the data because of their identical playout representations.

4.6 Principle 5 – UMID Inheritance

Any Instance derived from an original Material may be attached with a UMID composed of Material Number inherited from that of the original Material together with the non-zero Instance Number.

<u>Implications</u>

- A specific derivation method may vary depending on the application such as transcoding, partial retrieval, color correction, and so on, which is beyond the scope of these principles.
- The uniqueness of Instance Number may be guaranteed only within a predefined closed domain where the uniqueness of Instance Number is appropriately controlled.
- Instances with non-zero Instance Number but sharing the same Material Number shall be deemed to belong to the same group composed of Instances derived from a certain original Material as an anchor.
- Any Instance may be considered as original Material at its new entry, when a newly created UMID with a zero Instance Number replaces the existing one regardless of its creation history.

4.7 Principle 6 – Extended UMID

Any Extended UMID attached to a Material Unit belonging to the same material shall share the same Basic UMID that uniquely identifies the material as a whole.

Implications

- An extended UMID should be used to uniquely identify a finer granularity of material called the Material Unit.
- The Extended UMID may be used to uniquely identify the Material as a whole when its Material Unit is identical with the entire Material.

4.8 Principle 7 – Source Pack

The Source Pack should identify the time, space and ownership properties of the material at its point of creation or capture. If present as a part of an Extended UMID, its values shall not be replaced with new ones when the basic part of its UMID is inherited.

Implications

- When absent in an incoming essence file or stream, a newly created Source Pack may be appended if needed provided that meaningful and accurate values are determined.
- The Source Pack, if received as a part of the Extended UMID, may be replaced with new values at the point of entry of the Material when the basic part of its UMID is also newly created. However, doing this might then lose all reference to the origin of the Material and thus discouraged.
- The Source Pack, if received as a part of the Extended UMID, may be inherited as is regardless of the treatment of the Basic part of its UMID.

5 UMID-Aware Application Considerations

5.1 Introduction

This section describes several points on UMID usage that need to be carefully considered for UMID-aware applications in addition to the UMID Application Principles.

5.2 UMID Managed Domain

5.2.1 What is the UMID Managed Domain?

To embody the UMID Application Principles, a concept called UMID Managed Domain is introduced. The UMID Managed Domain is defined as an authoritative source or domain composed of materials, each of which is appropriately managed using UMID for its identity.

By definition, a UMID assigned to a material in a UMID Managed Domain shall be a valid UMID in accordance with the UMID Application Principles. Furthermore, because the UMID, by itself, is an identifier and not a locator, it contains no information for accessing the material it uniquely identifies. Therefore the information used to access the identified material, such as a URL (Uniform Resource Locator), shall be managed and associated with the UMID in the UMID Managed Domain. The UMID Managed Domain shall also provide the means by which a UMID is resolved to zero or more locators (e.g., absolute URIs and/or URLs), known as the UMID Resolution Protocol, which shall facilitate locating and accessing the identified material instances.

Figure 1 below illustrates a UMID Managed Domain. In this figure, “Material 1”, “Material 2”, “Material 3”, and “Material 4” are recorded on a certain physical medium as media files, while “Material 5” and “Material 6” are stored elsewhere on a separate medium. A Material Manager shall be provided to ensure the validity of UMID and to maintain information associated with UMIDs within the UMID Managed Domain. The Material Manager shall implement the UMID Resolution Protocol, shall provide a means to generate unique Material and Instance Numbers, shall provide a means to validate the uniqueness of UMIDs, and may contain any other descriptive metadata associated with the UMIDs within the UMID Managed Domain. The red arrows in Figure 1 from the Material Manager to the materials denote that the Material Manager recognizes the materials

("Materials 1", "Material 2", "Material 4" and "Material 5") as media files via their UMIDs (U_1, U_2, U_4 and U_5) and contains the URLs for the respective media files. Therefore, the Material Manager can provide the URL(s) for any of those materials in response to the UMID for one of them.

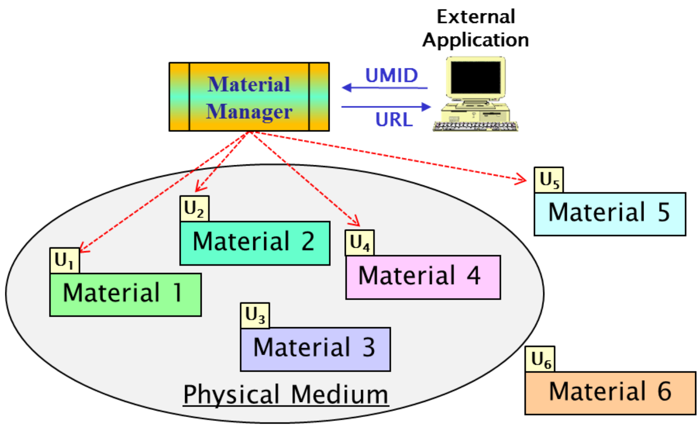

Figure 1 – UMID Managed Domain

Note that only the materials managed by the Material Manager (via their UMIDs) can constitute the UMID Managed Domain. In other words, materials that are not recognized by the Material Manager are out of scope of the UMID Managed Domain regardless of their location. As a result, even materials recorded on the same physical medium, such as "Material 3", cannot be a part of the same UMID Managed Domain, if they are not managed by the same Material Manager (*i.e.*, no red arrow from the Material Manager to the material in Figure 1) while materials recorded elsewhere, such as "Material 5", can be a part of the UMID Managed Domain because they are known to the Material Manager.

Also note that because external applications interact with the Media Manager to query information about the material it is managing, only information about material known to the Media Manager, including that for accessing the material uniquely identified by its UMID, can be returned (*i.e.*, material within the UMID Managed Domain). In other words, materials that are not recognized by the Material Manager *cannot be seen* from the external applications.

5.2.2 Material Manager Requirements

It shall be the responsibility of the Material Manager to maintain the integrity of the UMIDs that are assigned to the materials within the UMID Managed Domain that is managed by the Material Manager. Furthermore, to facilitate the UMID Resolution Protocol, it shall be also the Material Manager's responsibility to track the information required to access the material identified by each UMID within the UMID Managed Domain. Consequently, the material manager shall be involved in any kinds of material manipulation within the UMID Managed Domain.

Note: This does not mean that the Material Manager always needs to keep watching the UMID inside a media file on a recording medium. For a recording device with a removable recoding medium such as an optical disc as the UMID Managed Domain, the Material Manager installed in the device would be directly involved in re-writing a media file with a newly created UMID. For large media storage system composed of multiple storage devices, each of which constitutes the UMID Managed Domain, the Material Manager that governs the entire system would delegate actual UMID manipulation to the media file writer assigned to each storage device and request them to report when any change happens to the UMID and/or its associated access information.

While the detailed behaviors of the Material Manager to be implemented are application specific, any Material Manger shall take into account the following points:

- The Material Manager shall provide a globally unique UMID Material Number in accordance with SMPTE ST 330 upon demand.
- The Material Manager may provide a unique UMID Instance Number for a given Material Number upon demand.
- The Material Manger shall maintain the global uniqueness of UMID with zero Instance Number.
- The Material Manager may maintain the uniqueness of UMID with non-zero Instance Number for a given Material Number.
- The Material Manager shall implement the UMID Resolution Protocol, providing with zero or more information to access a material identified by a UMID in question.
- The Material Manager may store and retrieve descriptive metadata associated with unique UMIDs.

In Annex B, some typical behaviors the Material Manager will take in the case of a UMID as a globally unique material identifier for the material server equipped with rewritable recording medium are discussed in details.

5.3 UMID Material Number as a Dumb Number

The role of the UMID Material Number is nothing more than a theoretically proven globally unique value. Any meaning embedded in the UMID Material Number is accidental to the method of creating a globally unique value, and therefore, under all circumstances the value of the UMID Material Number shall be treated as a dumb number or a token; *i.e.*, the UMID Material Number value should not parse to obtain information from it.

Behind this requirement, on the other hand, an artificial modification of a field value in the UMID Material Number should be acceptable in order to create another theoretically proven globally unique value.

Suppose a situation where more than one material is generated simultaneously by a single recording device. According to the SMPTE Method for the UMID Material Number creation specified in SMPTE ST 330 Annex A, the UMID Material Number is created as a combination of 8-byte time snap value, 2-byte random number and 6-byte device node number as depicted in Figure A.1 of the Annex A. Therefore, when multiple materials are created simultaneously by a single recording device, such as in the case of stereoscopic 3D image or multichannel audio recording, there is a possibility that the same UMID Material Number will be assigned to those materials if the same value is generated more than once for the 2-byte random number.

To avoid such a problem, instead of using a random number generator, a method of assigning distinct values to the 2-byte random number field of each Material Number and/or duplicate prevention mechanisms to prevent the same random number from being generated more than once within the same unit of time may be implemented.

Furthermore, because the aforementioned time snap value itself is not significant (it is just for distinguishing the UMID Material Numbers created under a given device node number), its artificial modification should be also permitted. For example, if two UMID Material Numbers are required for a couple of materials simultaneously created by a single recording device, such a method is permitted that an original time snap value is used for the first Material Number creation while a time snap value intentionally incremented by one unit since the original value is used for the second Material Number creation in order to guarantee the distinction between those two UMID Material Number values.

Such an artificial treatment of the field values shall be acceptable only for the UMID Material Number creation in order for its value to be a theoretically proven globally unique dumb member. Because the date stamp and unit count component of the UMID Source Pack (a.k.a. the “When” component) is significant to identify the time and date of origin of the material unit with which the Source Pack is associated, the creation of the component value shall strictly follow what is specified in SMPTE ST 330.

Note: Even if an identical value is created for the “When” component of the UMID Source Pack over the simultaneously created multiple materials, a material unit attached with the Source Pack within a certain material is made distinct from that within another material by the Basic part of the Extended UMID attached to the material unit.

When materials of different material types are created simultaneously, their distinction via their UMIDs shall be made by specifying the correct material type to the 11th byte of the UMID (see Annex A.2). Because UMID is evaluated by the 32-byte long value as a whole, even if their UMIDs share the same UMID Material Number, materials of picture and sound material types simultaneously created by a single recording device can be made distinct with the 11th byte of their UMIDs as, *e.g.*, YCbCr picture type (06_h) and stereo sound type (09_h), respectively.

In general, no relationship shall be assumed among UMIDs sharing the same Material Number but have different material types. Its interpretation is left to each application’s discretion and thus beyond the scope of this Recommended Practice.

5.4 UMID as a Wrapper for the Globally Unique Value

As introduced in Annex A.2, the Basic UMID is composed of four fields called SMPTE Universal Label (UL), Length, Instance Number (Inst.#) and Material Number (Mat.#). When a UMID is used as a globally unique material identifier with zero Inst.# as shown in Figure C.1 in Annex C, the UMID is regarded as a *wrapper* of the 16-byte Mat.# which is globally unique on its own.

SMPTE ST 330 Annex A specifies four methods for the Mat.# creation. Because they are independent one another, there is a possibility that the Mat.# values created by different Mat.# creation methods collide. Thanks to the top nibble of the UL 12th byte of UMID specifying the Mat.# creation method, even the UMIDs that happen to have the same Mat.# value created by different Mat.# creation methods are made distinct.

Among those Mat.# creation methods, the most well-known one is the UUID method, which is commonly used in order to globally uniquely identify any entity in a generic information technology (IT) system.

Because no collision occurs as long as the 16-byte value is created by a single Mat.# creation method, the Mat.# created based on the UUID Method is guaranteed not to collide with any other UUID values created elsewhere. As a result, a material assigned with a UUID as a globally unique material identifier for the material to be managed in a global sense by following the convention of a generic IT system may be treated seamlessly with the materials having UMID of the UUID Method in principle because the difference between those unique material identifiers remains as a pure UUID or it is wrapped to form a 32-byte UMID.

Care should be taken, however, that how to treat UMID shall be strictly regulated by the UMID Application Principles specified in Section 4 in order to make it as, *e.g.*, a trusted globally unique material identifier.

In fact, an assumption behind the seamless treatment of materials described above is that not only the UUID Method based UMID but also the pure UUID are treated by strictly following the UMID Application Principles. In other words, because UUID itself is just a pure globally unique value whose treatment depends on an application, it also could be used as a content identifier that does not distinguish a high resolution material from its low resolution proxy sharing the same content, which cannot be mapped to the UMID that requires a strict distinction between them.

5.5 UMID for Abstract Original Material

While UMID has been originally intended to globally uniquely identify a physical material which is playable by itself, it may be also used to uniquely identify an abstract material that is not playable by itself but can be used to instantiate actual playable materials when the object oriented design approach is taken into consideration.

In this approach, an original material is regarded as an abstract class logically representing the material, from which any actual playable physical material is created in a similar way to an instance being instantiated from a class in the object oriented design. This abstract original material is in fact metadata that describes its common basic technical properties at its playout such as types of essence and duration. In addition, the basic descriptive properties commonly applied regardless of its instantiation, such as an originally created date, title, content description, and usage information, can be also included in the abstract original material if needed. Figure 2 below shows such a use of UMID.

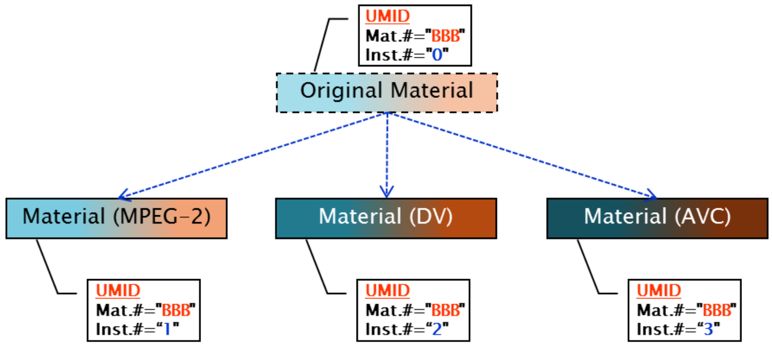

Figure 2 – Abstract Original Material and its Playable Instances

In this figure, the abstract original material is globally uniquely identified by a newly created UMID of zero Instance Number (Inst.#) while an actually playable physical material is attached with the UMID that inherits the Material Number (Mat.#) of the abstract original material and has the non-zero Inst.# depending on *e.g.*, the codec kind ("1", "2" and "3" for "MPEG-2", "DV" and "AVC", respectively).

Although those physical materials themselves are not identical according to the UMID Application Principle 4 (UMID Identification), they are known to share the same content because of their common parent class (the abstract original material). In other words, assuming the abstract original material as an original baseband signal for its ingestion, those physical materials are regarded as the ones specifically encoded from the original baseband signal, which are distinguished by its codec kinds specified as the Inst.# values within a closed domain where such an Inst.# assignment is defined elsewhere in advance.

Note: The Inst.# treatment described above is provided as an example only, which is effective only within a predefined closed domain fully controlled by the application, and thus beyond the scope of this Recommended Practice.

One of the benefits of this approach is a codec independent content edit. As the number of program delivery channels increases, the number of codec required for each of them will diverse. Because the abstract original materials in this approach are codec-free, an Edit Decision List (EDL) obtained from an edit of the originals (or, practically speaking, the EDL obtained from an edit of certain physical materials and their Inst.# being set to zero) is independent of any specific codec and therefore easily tailored to the EDL that meets the need of a specific delivery channel.

Note: Such an EDL instance itself can be also regarded as an abstract material if it is managed with the newly created UMID being attached to it.

Note, however, that because the UMID attached to a physical material is not a globally unique material identifier in this case, it cannot be used for the physical material to be managed in a global sense. In other words, if each physical material is also desired to be managed with its UMID in the same way as for the abstract original material, it also shall be attached with a newly created UMID of zero Inst.#, resulting in that the inheritance of the physical material from its abstract original material is no more described but should be supplemented by other external methods.

5.6 Identical Materials via UMID

According to the UMID Application Principle 4 (UMID Identification), materials whose playout results are identical may share the same UMID. To obtain better understanding of the meaning of this identicalness, Figure 3 below demonstrates an extreme case of three identical materials that can share a single UMID.

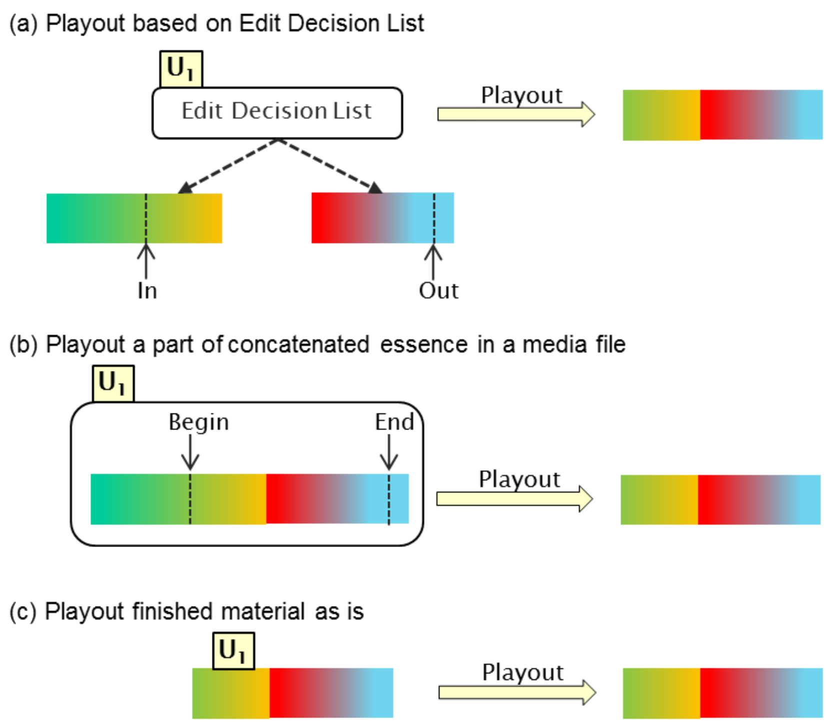

Figure 3 – Possible Identical Materials Sharing a Single UMID

Figure 3 (a) depicts a playout of EDL describing a concatenation of a part of existing materials specified by the In and Out points of the materials. Figure 3 (b) shows those existing materials being concatenated and stored into a file wrapper as an essence to form a media file, and a playout of a part of the inner essence by specifying its Begin and End timecodes corresponding to the In and Out points in Figure 3 (a), respectively, as is supported by certain media file formats such as MXF. Figure 3 (c) shows the finished material resulted from the execution of edit process specified in Figure 3 (a) or (b), and can be played out "as is".

Because (a), (b) and (c) in Figure 3 provide an identical representation at their playout by definition, they can share the same UMID according to the UMID Application Principle 4 (UMID Identification).

Note that the Principle does not exclude an attempt to assign different UMIDs to identical materials. If an application desires to distinguish and separately manage those three materials in Figure 3 in a global sense for some reasons, it may assign a newly created UMID of zero Instance Number to each of them.

5.7 UMID for Growing Material

When audiovisual material is created as a media file such as at the live recording, it is often hard to determine when the file creation is finished in advance. It is therefore desirable for the material being created to be made accessible even before its completion of creation, which is known as read-while-write operation, resulting in so called a growing material as shown in Figure 4 below.

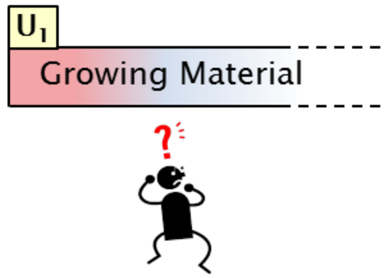

Figure 4 – Identification of Growing Material

It is important to note that UMID shall not be used to uniquely identify such a growing material in principle because it breaches the UMID Application Principle 3 (UMID Integrity); *i.e.*, as its duration continuously changes during growing, it is not deterministic and thus cannot be said unique.

Note: There is an experimental implementation of a growing material whose snapshot is frozen periodically and made accessible with a UMID newly created at every snapshot. Theoretically, it is valid because each frozen snapshot is deterministic and thus can be uniquely identified by a newly created UMID specifically for the snapshot.

On the other hand, a simple mechanism has been often desired to indicate that a particular material is under growing, for which the UMID with "Live stream" flag may be used.

The UMID "Live stream" flag is the value of F_h that is set to the bottom nibble of the byte 12^{th} of UMID (the field for Instance Number Generation Method). According to SMPTE ST 330, the UMID "Live stream" flag, or the value of F_h for the field, is specified as

> *"This instance number method shall be used to identify that the material is a direct live signal source from a material creation device",*

and, therefore, the UMID with the "Live stream" flag may be used to simply signal that a material attached with it is under growing rather than as a unique material identifier.

Note: While the UMID "Live stream" flag is originally intended for an un-persistent material from a device such as a camera and not yet being instantiated (*i.e.*, still remains "on the wire"), the use of the flag for the growing material is rationalized by the fact that initial part of material already stored on the recording medium seamlessly continues to the remaining part of material still on the wire, resulting in both of them to be regarded in the same status.

Note that the use of UMID "Live stream" flag for a growing material is just as a simple tool to signal the material growing, where only the value of F_h at the field for Instance Number Generation Method is significant regardless of the value specified at the rest of the fields in UMID. Because this use of UMID is not as a unique material identifier, it shall be replaced with a newly created UMID value at the completion of the material growing.

Annex A UMID Basics (Informative)

A.1 Introduction

This annex briefly introduces the basic information about UMID according to SMPTE ST 330, which will be helpful for understanding of the main body of this Recommended Practice, as well as notations to be used there.

A.2 UMID Format

The UMID is a byte string of either 32 or 64 bytes. The former is called the Basic UMID and is the core component of a unique material identifier. The latter is called the Extended UMID and is composed of two parts: 32-byte Basic UMID immediately followed by 32-byte Source Pack, as illustrated in Figure A.1 below.

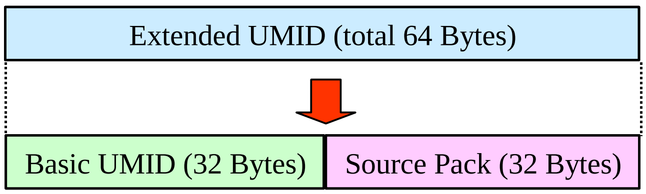

Figure A.1 – Top-level Structure of UMID

The UMID can be used either as the 32-byte Basic UMID on its own or as the 64-byte Extended UMID. Figure A.2 below shows the Basic UMID format, which is composed of the following four fields:

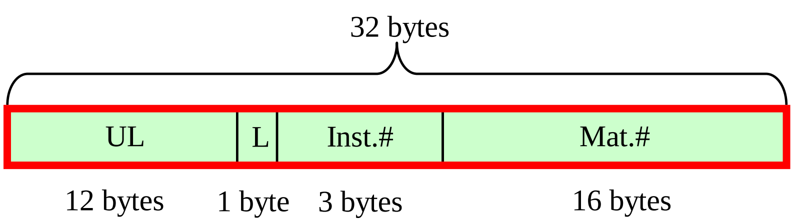

Figure A.2 – Basic UMID Format

- **SMPTE Universal Label (UL)**: The first 12 bytes of the Basic UMID constitute the SMPTE UL. The first 10 bytes are fixed values based on the registered International Organization for Standardization label administered by SMPTE. The 11th byte indicates the type of material this UMID identifies. The 12th byte is divided into top and bottom nibbles: the top nibble composed of 4 most significant bits (MSBs) and the bottom nibble of 4 least significant bits (LSBs) indicate the number creation methods for the Material Number (Mat.#) and Instance Number (Inst.#), respectively. Note that the bottom nibble of the 12th byte is also used to indicate the “Live stream”, *i.e.*, when this field is filled with the value of F_h, it indicates that a material attached with this UMID is a direct live signal source from a material creation device, resulting in being un-persistent, and thus cannot be uniquely identified,

  Note: According to SMPTE ST 298, the original SMPTE UL has a length of 16 bytes. This 12-byte SMPTE UL for UMID is created by truncating the last 4 bytes of zero value and changing the value of the label size in byte 2 from $0E_h$ to $0A_h$, which is still a valid and unique ISO object identifier defined in SMPTE ST 298.

- **Length (L)**: This 1-byte field specifies the length of byte string that follows. Since 19 bytes follow in the case of Basic UMID, this field is fixed to 13_h, while it is 33_h for the Extended UMID because of additional 32 byte Source Pack that immediately follows,

- **Instance Number (Inst.#)**: This 3-byte field specifies whether the Mat.#, the field that immediately follows this, is a newly created value or the inherited one from another UMID already existing elsewhere. For a newly created UMID, this field must be zero-filled (00_h 00_h 00_h), indicating that the Mat.# is a newly created value. While several methods are defined for the creation of non-zero Inst.#, the distinction of zero or non-zero value for this field is significant for most UMID applications,

- **Material Number (Mat.#)**: This 16-byte field accommodates a globally unique value, which makes the UMID also a globally unique material identifier. Several creation methods of a value for the field are specified, among which an example is given by a combination of the network node number of a device creating a material together with a time stamp at which the material is created. Because all network node number is globally unique, a material with UMID of this combination can be also globally uniquely identified under the assumption of a single material being created at a particular time. Note that this method creates a self-identifying globally unique value without access to either the registration authority or the central database, which is in common use in the information technology system such as for UUID (Universally Unique IDentifier).

When an individual material unit, or quantum duration of a material such as a video frame or an AES3 audio frame, needs to be uniquely identified within a material, an additional component called Source Pack is appended to the Basic UMID, which forms the Extended UMID as shown in Figure A.1.

Figure A.3 below shows the Extended UMID format, where the Source Pack is composed of the following three fields:

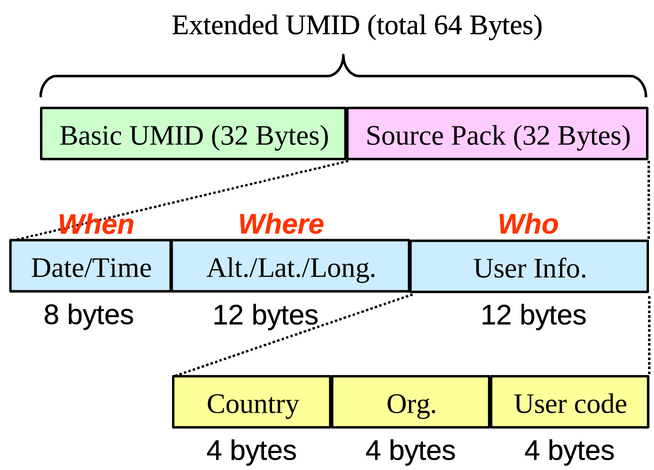

Figure A.3 – Extended UMID Format

- **"When" field (Date/Time)**: This 8-byte field specifies the date and time stamp at which a target material unit is initially created. The timing granularity for this field needs be smaller than the cyclic period of material unit so that each material unit can be uniquely identified with the value for this field.

- **"Where" field (Alt./Lat./Long.)**: This 12-byte field specifies the spatial co-ordinate information associated with a target material unit. This field is composed of three parts: Altitude, Latitude and Longitude, each of which is 4-bytes long.

- **"Who" Field (User Info)**: This 12-byte field specifies who initially created the target material unit as a string of pre-registered text. This field is composed of three parts: Country, Organization and User code, each of which is 4-bytes long.

While the original intention of the Extended UMID was to identify individual material unit that comprises the material, it also accommodates fundamental information about the creation of each material unit and thus needs to be preserved throughout the processes of the media production workflow chain.

Annex B To Maintain the UMID Managed Domain (Informative)

B.1 Introduction

This annex discusses some typical behaviors the material manager ought to take in order to maintain the UMID Managed Domain for the material server equipped with rewritable recording medium. Because only the primary function of UMID as a globally unique material identifier is considered, the Material Number (Mat.#) and the Instance Number are always set to a globally unique value and zero-filled, respectively.

B.2 Behaviors of Material Manager

The following describes the expected behaviors of the material manager for each material manipulation under the assumption that a material is recorded as a media file, whose URL is managed together with its corresponding UMID globally uniquely identifying the material by the material manager using the database called UMID Managed List.

B.2.1 New material creation in the domain

When a new material as a media file is created from scratch by, *e.g.*, acquisition, in the domain, the material manager will assign a newly created UMID to the media file, and also register the pair of the UMID and the file's URL to the UMID Managed List.

Note that if the media file is accessible even during at its creation (the growing media file), a.k.a. the read-while-write operation, the UMID attached to the media file will indicate the "Live stream" by setting the bottom nibble of the byte 12^{th} of UMID to F_h. This is because such a growing media file is un-persistent and thus cannot be uniquely identified. As a result, the UMID with the F_h value cannot be regarded as a globally unique material identifier.

When recording stops, resulting in the media file being persistent and thus can be globally uniquely identified, its UMID will be made valid by completely replacing the UMID with a newly created value having zero Inst.#.

The registration of the UMID and URL pair to the UMID Managed List is required only when the media file becomes persistent with a newly created UMID as its valid globally unique material identifier.

B.2.2 Existing material deletion in the domain

When an existing material as a media file is deleted in the domain, the material manager will retain a URL of the file being deleted, detect the UMID and URL pair in the UMID Managed List according to the URL as a key, and unregister it from the UMID Managed List before the file deletion. Note that while there might exist more than one clone of a media file in the domain which shares the same UMID value as that of the media file being deleted, the use of the URL as a key can unambiguously identify the UMID and URL pair to be unregistered from the List.

Note: The un-registration of the UMID and URL pair from the UMID Managed List eventually results in its deletion from the viewpoint of an external application that will request the material manager for the UMID resolution regardless whether the media file is actually deleted or not.

B.2.3 Existing material copy or move within the domain

When a new material as a media file is created by copying an existing source media file within the domain, the material manager will detect the UMID of a source media file, assign it also to the copied media file, and register the pair of the UMID and the URL of a copied file to the UMID Managed List.

Note: In practice, this behavior can be reduced to duplicating the UMID and URL pair of the source media file within the UMID Managed List and updating the URL of one of the pairs to the value of a copied media file after the file copy operation.

If this is a move operation where the source media file in the domain is deleted after the copy operation, the behavior for the existing material deletion will apply to the source media file.

Note: If the operation is known as the move in advance, a practical behavior of the material manager will be just to update the URL value to that of destination of the media file being moved.

It ought to be noted that in either case, the material manager might intentionally replace the UMID of resulting material with a newly created value for some reasons, which is still valid in the sense of the UMID Application Principles. While such a possibility cannot be excluded, its further discussion is beyond the scope of this Annex.

Note: If the copy operation fails and results in leaving an incomplete media file, the resulting media file must be assigned with a newly created UMID to unambiguously distinguish it from its source media file. If such an incomplete media file is still to be used rather than to be deleted for the rollback, a practical behavior the material manager will take is to tentatively assign a newly created UMID (or the UMID with a “Live stream” flag when the read-while-write operation is permitted) to the resulting media file at the beginning of the copy or move operation and then to replace the tentative UMID with that of its source media file at the successful completion of the operation.

B.2.4 New material import into the domain

When a new material as a media file is imported to the domain by copying or moving from elsewhere, the default behavior the material manager will take includes to detect the UMID of an incoming media file, to replace it with a newly created UMID value if detected (or to simply assign a newly created UMID to the media file if not detected), and to register the pair of the UMID and the resulting file’s URL in the domain to the UMID Managed List.

Note: This is also applicable to a material existing on the same physical medium but not recognized by the material manager such as “Material 3” in Figure 2 when it is to be imported into the UMID Managed Domain.

Note: This implies that the value of original UMID is lost in the import operation in general. Hence, if the original UMID is desired to be retained, an external database that records the UMID tracking needs to be provided. In the case of MXF, however, thanks to the layered structure of the MXF Packages, there is a way for such an original UMID to be retained even in the imported material as an MXF file.

An exceptional behavior can be applied when the location from which a media file is to be imported is also known as the UMID Managed Domain. In this case, because the UMID attached to the incoming media file is valid in the sense of the UMID Application Principles by definition, the UMID can be reused when a media file is imported into the domain exactly “as is” (as a clone of the media file). Therefore the material manager in this case will just detect the UMID of an incoming media file, and then register the pair of the UMID and the resulting file’s URL in the domain to the UMID Managed List.

Note: According to the UMID Application Principles 4 (UMID Identification), identical materials can share the same UMID. Consequently, this behavior is applicable to the material import not only as a clone but also as an identical one in the sense of the UMID Application Principles such as just for its file wrapper or its descriptive metadata being changed.

Care needs to be taken that this behavior is effective only when the detected UMID is deemed as a valid globally unique material identifier with a zero Inst.#. In other words, if the detected UMID has non-zero Inst.# or the UMID “Live stream” flag (the bottom nibble of the byte 12^{th} of UMID being set to F_h) resulted from a copy of a growing media file, the default behavior, or to replace it with a newly created UMID value, will apply because of their invalidity as a globally unique material identifier.

Note: Even when a material with a valid UMID is imported from the UMID Managed Domain, a material being recorded in the domain will signal its status by specifying the UMID “Live stream” flag as the F_h value if the material is made accessible before completion of its recording. As a result, even if the file import is discontinued before the completion for some reasons, the incompleteness of the remaining file in the domain can be detected by its UMID with the flag, which needs to be replaced with a newly created UMID value when the incompletely recorded media file is still to be used in the UMID Managed Domain.

It ought to be noted again that, even though a media file is known to come from another UMID Managed Domain in advance, the material manager might intentionally replace the incoming UMID with a newly created UMID value for some reasons, which is still valid in the sense of the UMID Application Principles. While such a possibility cannot be excluded, its further discussion is again beyond the scope of this Annex.

B.2.5 Existing material export from the domain

When an existing material as a media file in the domain is exported to outside of the domain, no particular behavior is required by the material manager as long as the source material remains as is in the domain. It is the responsibility of the material manager in destination (if exists) to guarantee the validity of the UMID. If this is a move operation where the source media file in the domain is deleted after the export, the behavior for the existing material deletion will apply to the source media file.

Note that if the destination of the material export is not the UMID Managed Domain, the material is thus no more managed according to its UMID. As a result, the material at the destination cannot be seen from an external application that will request the material manager for its UMID resolution to its corresponding URL.

B.2.6 Existing material modification at its essence in the domain

When an existing material as a media file in the domain is modified at its essence by, for example, insert editing, the material manager will find the UMID of the media file, replace it with a newly created UMID value, and update the UMID value in the pair of the UMID and the file's URL existing in the UMID Managed List.

This needs to be conducted even when only one media file is known to exist for a given UMID within the domain, *i.e.*, because the scope of uniqueness of UMID is global, we always need to assume an existence of identical material sharing the same UMID elsewhere unknown. Hence, in order to meet the UMID Application Principle 3 (UMID Integrity), it is mandatory for the material manager to update the UMID at any essence modification except for the essence manipulations that logically and fully preserve the identicalness of the material such as mathematically lossless compression.

Note: Although no action seems needed at the material modification to meet the UMID Application Principle 3 (UMID Integrity) if nonexistence of the identical material elsewhere is logically proven, it does not meet the UMID Application Principle 4 (UMID Identification) at all. This would be better understood by such an extreme example that when the essence in a media file is completely replaced with another essence, it will make sense for its UMID to be also replaced with a newly created UMID value.

It ought to be noted that any UMID update at its essence modification will break the link to the original media file from any external entity elsewhere, such as external metadata. Hence it is an application's responsibility for the resulting media file to be re-linked to the external entity if needed such as for a description irrelevant to the essence.

B.2.7 Existing material modification at its metadata in the domain

When an existing material as a media file in the domain is modified at its metadata, the behavior the material manager will take depends on the types of metadata being modified.

If the metadata is descriptive metadata such as title or content synopsis, there is nothing for the material manager to take because it does not affect the material representation at its playout.

If the metadata is structural metadata based on which the material representation at its playout is controlled, the material manager will detect and update the UMID to a newly created UMID value in order to meet the UMID Application Principle 3 (UMID Integrity).

B.2.8 Existing material backup and restore

When an existing material as a media file in the domain is stored into another storage medium for backup, the material manager will take no action even though the materials in the storage medium become out of scope of the manager. This is because such a storage medium will be regarded as an extended area of the domain, where materials in the storage medium are regarded as equivalent with those in the domain and thus their UMID Integrity is guaranteed.

Hence, when a backed up material in the storage medium is to be restored to the original domain, the material manager will take the same behavior as for the import from another UMID Managed Domain.

It ought to be noted that this is based on the assumption that the backup is regarded as an operation to store and to restore a clone of a material, and thus, if only a part of the backed up material is restored to the domain by so called partial retrieval operation, this assumption does not hold and the behavior for the import from somewhere unknown will apply.

B.2.9 Material managed by multiple material managers

Theoretically, a single material as a media file can be managed by more than one material manager. Because UMID attached to a material in the UMID Managed Domain is valid, another material manager can take such an action that it does not actually import the material into its own domain but just references to the material and registers the UMID and URL pair for the material to its UMID Managed List.

Because each material manager basically functions independently, a problem occurs when a material manager is involved in a material manipulation that will change the state of the material. For example, when one material manager is involved in a material deletion, the manager would unregister the UMID and URL pair for the material in its UMID Managed List. But because the pair in the UMID Managed List for another material manager remains unless appropriately notified with the deletion, the manager will respond to an external application with the URL which is no more effective.

More serious problem occurs when a material manager is involved in a modification of a material at its essence. Although one material manager involved in the modification will update the UMID attached to the material, it could happen that the UMID update is not reflected to the UMID Managed List of another material manager in a timely manner, resulting in that the manager continues to resolve the old UMID to the URL at which the original material identified by the old UMID does no more exist but modified one with the updated UMID.

Ideally, the material manager is expected to periodically examine the validity of UMID in its UMID Managed List by inspecting all the UMIDs attached to the materials in the domain. In reality, however, it would be too cumbersome for the manager to check the actual UMID for the material at all times.

To address this problem, one possible solution will be to compare the time stamp attached to the material at its modification with that attached to the UMID and URL pair at its registration to the UMID Managed List.

If we make it a rule to register the UMID and URL pair to the UMID Managed List *after* the completion of the material modification, the time stamp for the UMID and URL pair registration is usually newer than that for the material modification. Consequently, if a material manager detects the former time stamp being older than the latter, the UMID in the UMID Managed List of the manager might be obsolete because it indicates another material modification occurs after the UMID and URL pair registration to the UMID Managed List, which might change the state of the material and thus update the UMID.

This problem occurs only for the material manipulations that will change the state of the material and thus the UMID such as the material modification at its essence. Hence, we can avoid the problem if we prohibit such manipulations but just permit the read-only operation that will never change the state of the material.

Therefore, if you want to manage media files that might exist in other UMID Managed Domains (such as those in another independent storage device), it would be better for you to treat it as a read-only file; *i.e.*, you ought not to delete or even slightly modify the essence in the media files in order to avoid inconsistency in the UMID Managed List of other material managers.

Annex C UMID Application Examples (Informative)

C.1 Introduction

This annex introduces various examples of UMID applications collected during the study of the UMID applications that has produced this Recommended Practice. It ought to be noted that while an original intention for this annex was to provide the “best practice" of UMID applications, there is no unambiguous criterion to determine which are the best. Hence, as long as they are valid UMID applications in the sense of the UMID Application Principles, they are incorporated into in this annex.

Note that the UMID applications discussed in this annex are generic; *i.e.*, since no assumption is made for the material as a media file of certain format, they can be applicable to any kinds of representation of the material.

C.2 UMID Application Basics

C.2.1 Two distinct functions of UMID

The primary function of UMID is as a globally unique material identifier (called Application 1, hereafter). When a new audiovisual (AV) material is created from scratch by *e.g.*, acquisition, a new UMID is created and attached to the material automatically so that it can be globally uniquely identified by the UMID value. In this case, while the Material Number (Mat.#) for the UMID is a newly created globally unique value, the Instance Number (Inst.#) must be zero-filled as show in Figure C.1 below, indicating that it is the original AV material.

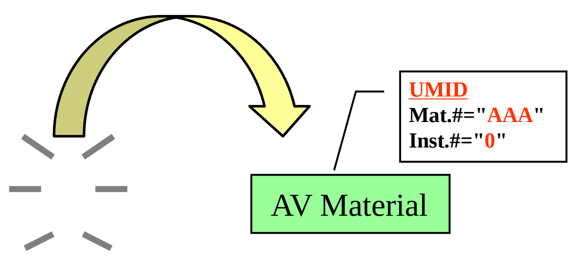

Figure C.1 – UMID as a Globally Unique Material Identifier (Application 1)

Note that it is assumed in the figure that the material being created cannot be accessed until completion of its creation together with the UMID attachment. If any material is made accessible before the completion such as for a recording of live media stream (the growing media file), a.k.a. the read-while-write operation, the material has either no UMID or a UMID with the “Live stream” flag being set, *i.e.*, the value of F_h for the bottom nibble of the byte 12^{th} of UMID.

Another function of UMID is as a linking tool (called Application 2, hereafter). When a new AV material is created from existing source material by *e.g.*, partial retrieval, a UMID whose Mat.# being inherited from that of the source material while Inst.# set to non-zero will be attached to the resulting material as shown in Figure C.2 below.

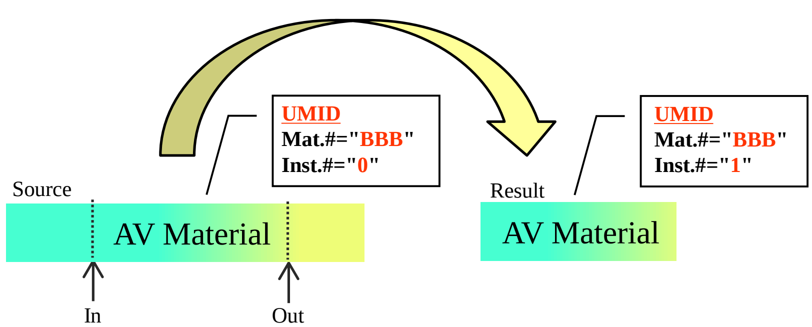

Figure C.2 – UMID as a Linking Tool (Application 2)

In this case, the resulting material is logically associated with the source material via their Mat.#; *i.e.*, the UMID of the source material can be easily obtained by just masking the Inst.# of UMID for the resulting material to zero.

While partial retrieval is taken up here for the example of UMID as a linking tool, any kinds of material derived from the source material can have such a UMID, which include, but are not limited to, material transcoded into another codec, material having different file wrapper or metadata, or a thumbnail image extracted from the source material.

C.2.2 Exclusive UMID functions

While there are two functions of UMID as a globally unique material identifier (Application 1) and as a linking tool (Application 2), those two functions are completely exclusive. If a UMID is used as a linking tool as is demonstrated in Figure C.2, it can no more be a guaranteed globally unique material identifier.

This is because the value space of the Inst.# of 3-byte long is far from sufficient to accommodate a globally unique value. While it would be possible for an application to carefully control the Inst.# so that the UMID remains unique, its uniqueness is guaranteed only within a predefined closed domain in which the application can fully control the uniqueness of the Inst.# value. In other words, it could happen that two independent applications create the truncated materials of different durations from the same source material but happen to assign the same Inst.# value, resulting in breaking the global uniqueness of UMID.

This leads to the conclusion that when a material at hand has a UMID with a non-zero Inst.#, the material cannot be managed by the UMID in a global sense. Therefore if a material is desired to be managed by its UMID in the same way as for the originally created material shown in Figure C.1, the material needs to be assigned (or to replace its existing UMID) with a newly created UMID of zero Inst.# as a guaranteed globally unique material identifier.

C.2.3 Overview of UMID treatments in a media production workflow chain

As introduced in Annex C.2.1, there are two distinct functions of UMID, *i.e.*, as a globally unique material identifier (Application 1) and as a linking tool (Application 2). Taking account of those two basic UMID functions, an overview of the UMID treatments in a media production workflow chain is schematically illustrated in Figure C.3 below.

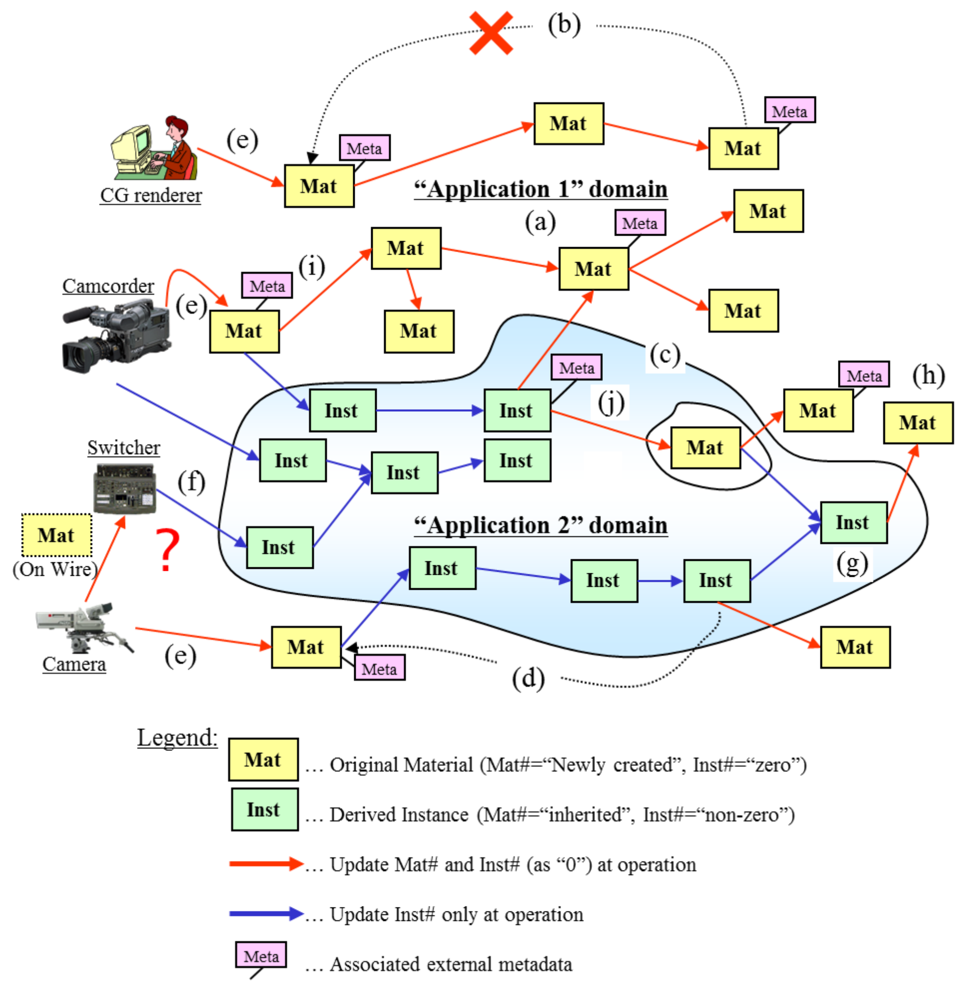

Figure C.3 – UMID Treatment in Media Production Workflow Chain

In this figure, the yellow box with "Mat" label represents a specific item of material with a UMID having a newly created Material Number (Mat.#) and zero Instance Number (Inst.#), while the light green box with "Inst" label represents the case of inherited Mat.# with non-zero Inst.#.

The red and blue arrows in the figure indicate operations causing the UMID updates to the new UMID (newly created Mat.# and zero Inst.#) and to the non-zero Inst.# only, respectively.

Two domains of UMID Applications are defined in the figure – "Application 1" domain composed of materials obtained only from the UMID as a globally unique material identifier (Application 1), and "Application 2" domain composed of materials obtained from either the Application 1 or the UMID as a linking tool (Application 2).

Note also that some materials are associated with external metadata, which is represented by the attached light purple box with "Meta" label.

The alphanumerical labels shown in Figure C.3 indicate as follows:

(a) Application 1 is applicable anywhere, including even within the Application 2 domain as a new entry (see the item (c) below) because it guarantees all the UMIDs in the domain to be globally unique by their newly created Mat.# and zero Inst.#.

(b) No link to a source material is established in Application 1, and thus if it is required it must be provided by other external methods.

(c) Application 2 is applicable to a certain closed domain in which the numbering of UMID Inst.# is appropriately managed so that the Inst.# is known to be unique for a given Mat.#.

(d) A link to a source material is always established in Application 2.

(e) A material instantiated as a persistent and deterministic form by a CG (Computer Graphics) renderer, a camcorder, or a baseband signal ingestion is deemed as an original material and hence must be attached with a newly created UMID of zero Inst.#.

(f) Theoretically, it would be possible that one regards a bounded sequence of video signal created from a device such as a camera and not yet instantiated (*i.e.*, it exists only "on the wire") as an original material and thus the UMID of inherited Mat.# and non-zero Inst.# is to be attached when recorded. Without an original material existing in a persistent and deterministic fashion, the usefulness of such a UMID is very much limited and thus this sort of UMID use is strongly discouraged.

(g) In Application 2, a new material composed of more than one source material can inherit the Mat.# of only one of the source materials and thus the decision as to which will be application dependent — perhaps, for example, according to relative data size, etc.

(h) Any UMID created as a result of material leaving the Application 2 closed domain must be a new value with zero Inst.# in order to guarantee its global uniqueness. This is equivalent with a new entry of the material into the Application 1 domain as an original material (See the item (e) above).

(i) A material deemed as the original can be associated with metadata that exists anywhere in the world.

(j) Any material derived from an original material can be associated with metadata that exists within a predefined closed domain where the local uniqueness of Inst.# is appropriately controlled. Note that metadata attached to such a material can be also deemed to be associated with its original material via its Mat.# (See the item (d) above).

C.3 Simultaneous Multiple Material Creations for Offline/Online Edits

In general, an original material is huge in its data size especially for high resolution. It is therefore common in practice for its lightweight proxy material to be used for its content viewing and so called offline edit instead of the original one. Because of its role, it is preferable for the proxy material to be tightly associated with its original one without ambiguity. To achieve this goal, UMID attached to the proxy material can be used as a linking tool to its original one.

Thanks to the increase of computational power, a modern capture device can generate more than one type of material simultaneously. A typical use of such a device will create high resolution material as an original one and its low-resolution version as the proxy at the same time. In fact, many commercially available camcorders have already supported such functions, resulting in simultaneous creations of the original and its proxy materials as schematically illustrated in Figure C.4 below.

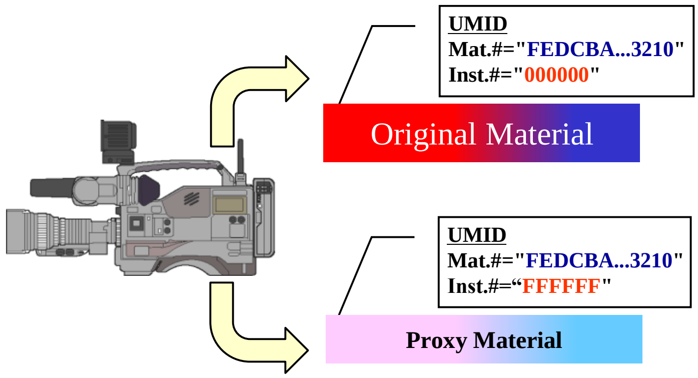

Figure C.4 – Simultaneous Creations of Original and Proxy Materials at Acquisition

Note that, in this figure, while a UMID as a globally unique material identifier is attached to the original high resolution material, a UMID as a linking tool to the original is attached to the proxy. With such materials, the EDL (Edit Decision List) obtained from the offline edit using the proxy can be easily translated to the EDL to be applied to the original ones for the online. For example, suppose a clipping of the material shown in Figure C.4 be requested with its timecodes from 00:01:20:00 to 00:05:00:00. An EDL obtained from the offline edit would be described by using the SMIL (Synchronized Multimedia Integration Language) like syntax as

```
<smil ...>
   <body>
      <par>
         <ref src="urn:smpte:umid:060A2B34...0D12.13FFFFFF.FEDCBA...3210"
            begin="00:00:00:00"
            clipBegin="00:01:20:00" clipEnd="00:05:00:00" ...
         />
      </par>
   </body>
</smil>
```

where the `ref` element describes the reference of an external material to be played out, the `src` attribute identifies the external material by its URN (Uniform Resource Name) while the `clipBegin` and `clipEnd` attributes specify the In and Out points for the clipping, respectively. Note that, for the URN value of the `src` attribute, the UMID of the proxy is specified in which `FFFFFF` in green font color corresponds to its Inst.#, immediately followed by its Mat.# as `FEDCBA...3210`.

An application of the EDL to the original material is straightforward, *i.e.*, because the difference of UMIDs for the original material and the proxy is their Inst.# only, just masking the Inst.# of the proxy to zero gives the EDL to be applied to the original material as

```
<smil ...>
   <body>
      <par>
         <ref src="urn:smpte:umid:060A2B34...0D12.13000000.FEDCBA...3210"
            begin="00:00:00:00"
            clipBegin="00:01:20:00" clipEnd="00:05:00:00" ...
         />
      </par>
   </body>
</smil>
```

Since the location of the original material is expected be found by the UMID resolution applied to the `src` value, a requested clipping of the original material is to be obtained when this EDL is executed.

It is worthwhile noting that while there are various ways of the EDL mapping between the offline and online edits, most of them are proprietary and thus can be conducted only within the pre-designated systems and/or applications. Thanks to the UMID as the SMPTE standard technology, however, such a use of UMID for the EDL mapping between the offline and online edits as demonstrated here can be conducted among applications from different vendors provided a common EDL syntax and the use of UMID for the proxy are shared among them in advance.

It needs to be also noted that because the UMID attached to the proxy material is not a globally unique material identifier, it cannot be used for the proxy to be managed in a global sense. In other words, if the proxy is also desired to be managed with its UMID in the same way as for the original high resolution material, it needs be also attached with a newly created UMID of zero Inst.#, resulting in that the UMID based EDL mapping demonstrated here cannot be achieved but ought to be supplemented by other external methods.

C.4 UMID based Material Search

One of the primary roles of metadata is its use for the audiovisual material search. Because of huge difference in data sizes between the material and its associated metadata, it is reasonable for the metadata to be stored separately from the material, and it is UMID that logically associate the metadata with the material.

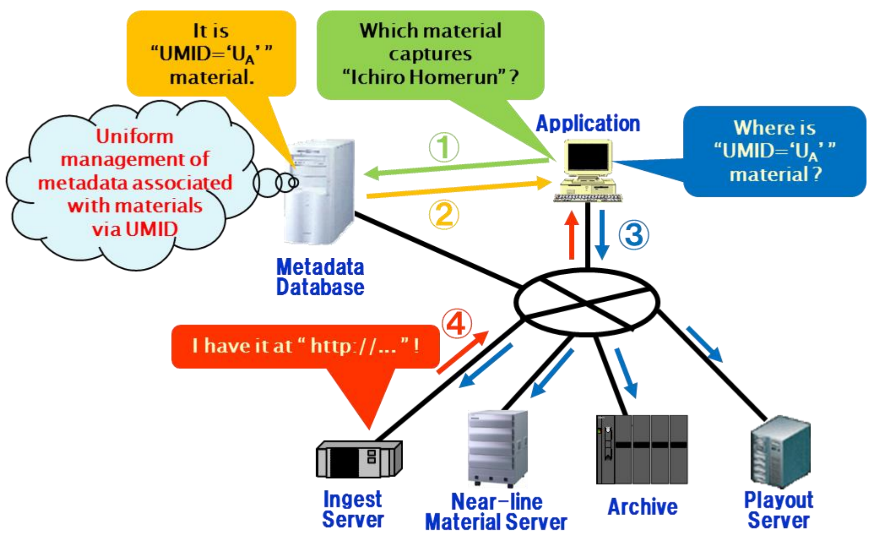

Figure C.5 – UMID based Material Search

Figure C.5 above shows a typical example of such a UMID application scenario, which is called the UMID based material search. In this scenario, audiovisual materials are stored in various kinds of material servers ("Ingest Server", "Near-line Material Server", "Archive", and "Playout Server") connected to the network in a media production system and metadata associated with a material via its UMID is collected and separately stored altogether in the dedicated metadata database ("Metadata Database") for their uniform management.

When an external application ("Application") such as video editing desires to obtain an audiovisual material that captures an "Ichiro Homerun" scene for example, it will gives a query to the metadata database accordingly. The metadata database will then reply to the application with the desired material by its UMID.

Because the UMID itself cannot tell anything about where to access the desired material, the application needs to resolve the UMID, *i.e.*, to convert the UMID into its corresponding URL for example. In this scenario, the application will distribute a query asking "Where is 'UMID=U_A' material?" and the Ingest Server ("Ingest

Server") will respond to it with the URL for the desired material, which is then to be used for the application to actually access the material.

Note that, in this application scenario, it is assumed that all the materials stored in any material server are appropriately managed based on their UMIDs as globally unique material identifiers within each server that strictly follows the UMID Application Principles, resulting in that each material is globally uniquely identified by its UMID regardless of the material server storing it.

C.5 Cooperation of Multiple MAMS (Media Asset Management System)

It is a common practice for an audiovisual material to be assigned with a unique material identifier in a media asset management system (MAMS) in order for the material to be managed consistently within the system. Usually, the scheme of the unique material identifier is proprietary to the system, which is effective only within the system. As a result, a problem occurs if a material is shared by more than one MAMS, which is also demonstrated by Figure C.6 (a) below.

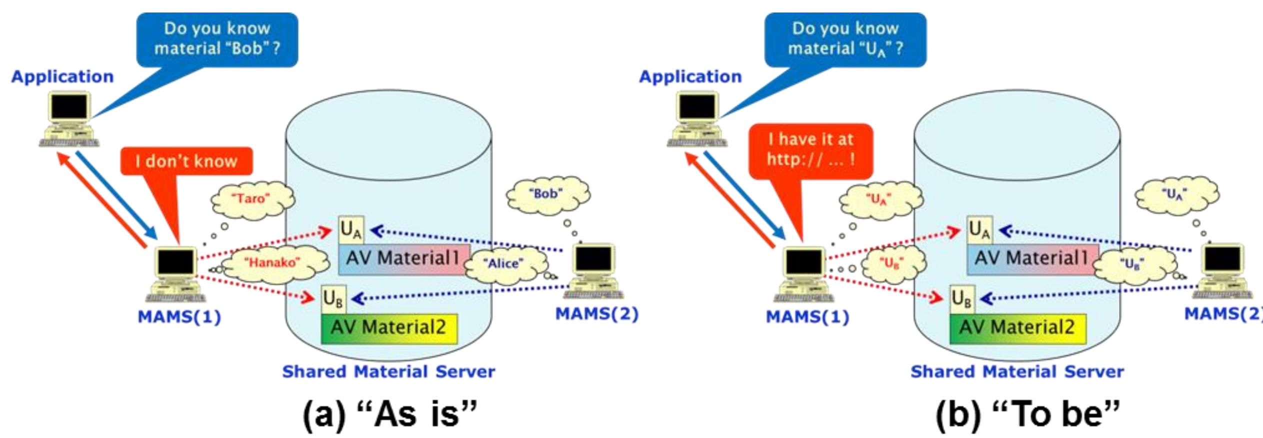

Figure C.6 – Cooperation of Multiple MAMs

In this figure, two MAMSs ("MAMS (1)" and "MAMS (2)") are provided, and two audiovisual materials ("AV Material1" and "AV Material2") stored in a shared material server ("Shared Material Server") are managed concurrently by both of them.

Because those MAMSs are independent each other, they would assign their own proprietary unique material identifiers to the materials. For example, "MAMS (1)" assigns the Japanese naming-like unique material identifiers such as "Taro" and "Hanako" to the materials while "MAMS (2)" the English naming-like ones such as "Alice" and "Bob".

Suppose there is an external application ("Application") that happens to ask "MAMS (1)" for the material "Bob". Because the identifier "Bob" is not a part of the material identification scheme for "MAMS (1)", it cannot but just respond to the application that "I don't know". So what will it be?

Because UMID is an industry standardized globally unique material identifier, a material in a shared material server is expected to be attached with UMID at its creation, which is independent of any specific products. Therefore, if each product, say "MAMS (1)" and "MAMS (2)" in this case, respects the UMID attached to the material and treats it as one of metadata items associated with the material, the application then can ask whichever MAMSs for the material by its UMID and either of them can correctly answer for it, as demonstrated by Figure C.6 (b).

It ought to be noted that this does not request a product to replace its own proprietary unique material identifier (which would be usually its primary key for a material it manages) with UMID. What is requested for the product is just to manage the UMID as one of metadata items associated with a material it manages, while care ought to be taken for the UMID to be always maintained valid in the sense of the UMID Application Principles.

Considering the UMID as an industry common primary key for the material search, it also works as an industry common standard interface for the material management product such as MAMS. Because such products are usually differentiated by their specific metadata support, an intensive discussion would be required to agree on the level of interoperability of the query for a desired material by its metadata. But there is no doubt that UMID is the minimum level of such interoperability. Moreover, because it is eventually the information to access a desired material which is required in response to the query for the material, the UMID resolution to such information as a URL for the material is indispensable.

Therefore, the UMID and its resolution will constitute the bottom-line of the interoperability among products for the material management and search.

C.6 Common Material Identifier in a Best-of-Breed Media Production System

Any media device that participates in a media production system and manipulates material in any way will have more or less a capability of the material management. If such a device manages the material based on their UMIDs and thus strictly follows the UMID Application Principles, it is also regarded to implement the UMID Managed Domain discussed in Section 5.2.

Because any material in the UMID Managed Domain is globally uniquely identified by definition, a material in a media production system is also globally uniquely identified by its UMID if the system is composed only of media devices supporting the UMID Managed Domain regardless of kind or vendor of the devices. In other words, as long as the media devices conforming to the UMID Application Principles are used to build the system, the uniqueness of UMID of any material in the system is perfectly guaranteed even when the devices are coming from multiple vendors, *i.e.*, known to as the best-of-breed system.

Figure C.7 below schematically illustrates such a media production system. In this figure, the UMID Managed Domain supported by a media device is represented as a background light green oval and the overlaps among the ovals express the marge of the UMID Managed Domains. As shown in the figure, when all the UMID Managed Domains supported by media devices participating in the system are merged, it will cover the entire system, resulting in the UMID being used as a unique material identifier over the system.

Note that, in this application scenario, it is assumed that every media device in the system appropriately supports the UMID resolution protocol, or the industry common method to convert a given UMID into its corresponding URL. As a consequence, any media device in the system can make materials it manages to be freely accessed from other devices and/or can access any materials stored elsewhere in the system.

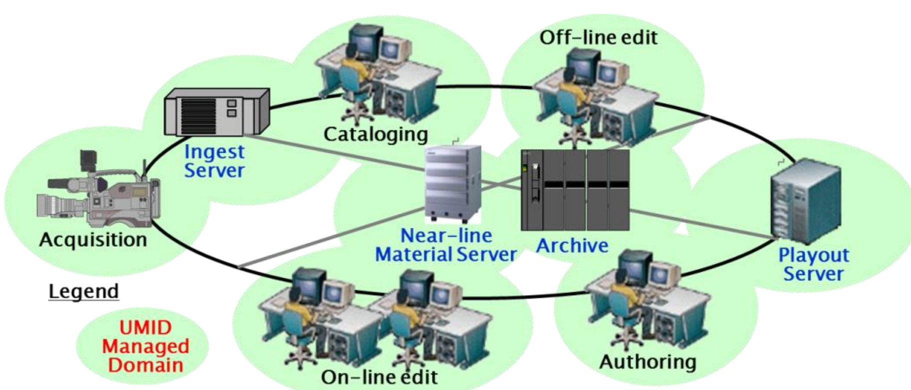

Figure C.7 – Media Production System using UMID as a Unique Material Identifier

C.7 Globally Distributed Material Management

The global uniqueness of UMID further allows for the application scenario described in Annex C.4 to be extended in a global sense regardless of geographical position of the media devices or even systems in the world. Figure C.8 below demonstrates such an application scenario.

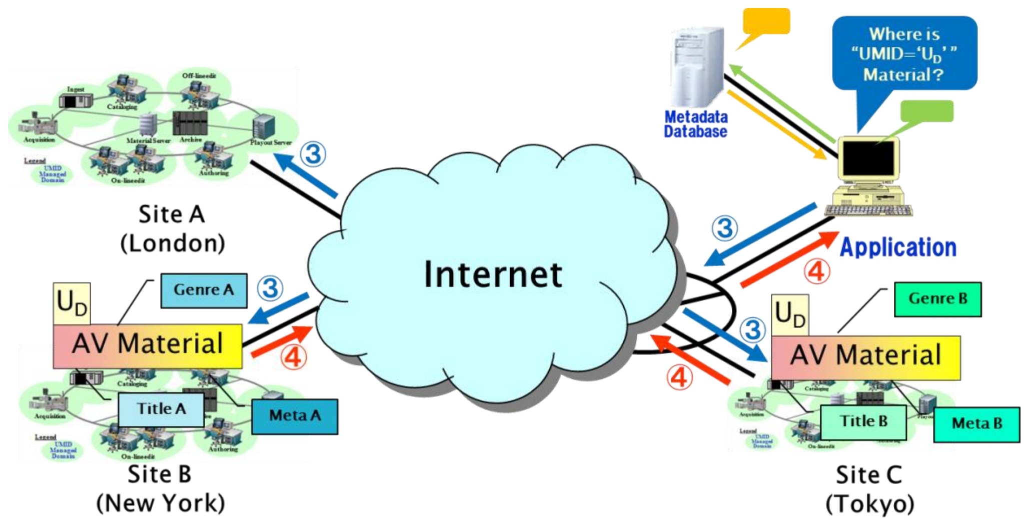

Figure C.8 – Globally Distributed Material Management

In this figure, the media productions systems strictly following the UMID Application Principles are located in London ("Site A"), New York ("Site B") and Tokyo ("Site C"). An external application ("Application") is assumed to desire a certain audiovisual material, to negotiate the metadata database and to obtain its UMID as 'U_D'. The application will then give a query asking "Where is 'UMID=U_D material?' " to all the sites via the Internet and obtain the information that the desired materials exist both in New York and Tokyo.

Note that because of cultural and/or language difference or whatsoever, the descriptive metadata associated with a material such as title and genre would change depending on the site. Furthermore, the material management system such as MAMS introduced into each site would differ depending on its business requirements. But if a media production system at each site strictly follows the UMID Application Principles, the UMID attached to any material in the system at any site is guaranteed to be globally unique.

In addition, thanks to the UMID Application Principle 4 (UMID Identification), the materials found in New York ("Site B") and Tokyo ("Site C") are guaranteed to be identical (in the sense of the representations at their playout according to the Principle 4) regardless of metadata associated with respective materials.

Therefore the application can access whichever it desires by taking account of other factors. As for the scenario in Figure C.8, it is likely that the application will access the material found in Tokyo ("Site C") because of its existence in the same sub-network.

In general, materials are huge in their data size and therefore it will be more preferable for the material movement over the network to be minimized, *i.e.*, ideally, all the materials are hopefully managed at the site of their initial creation and only a desired material or even a desired portion of a material is transferred to other sites over the network. On the other hand, metadata can be easily moved over the network because of its tiny data size, and thus it ought not to be a big burden for the metadata generated at all sites to be collected to a certain dedicated metadata database and to be managed in a uniform fashion.

It ought to be noted that the existing practices of material management so far cannot but either rely on the central material management system or be built upon only products from a single vendor, which is far from the ideal material management scenario described above.

If all the media devices and systems strictly follow the UMID Application Principles, however, the globally distributed material management, or the aforementioned ideal application scenario, can be realized based on the UMID as a globally unique material identifier as is clearly demonstrated in Figure C.8.

C.8 Additional Metadata Search for a Material at Hand

So far, the material search by UMID as a key has been mainly discussed. But there is a case where additional metadata associated with a material is desired to search by its UMID. One of such cases is to search for the terms and conditions of a material at hand, as demonstrated by Figure C.9 below.

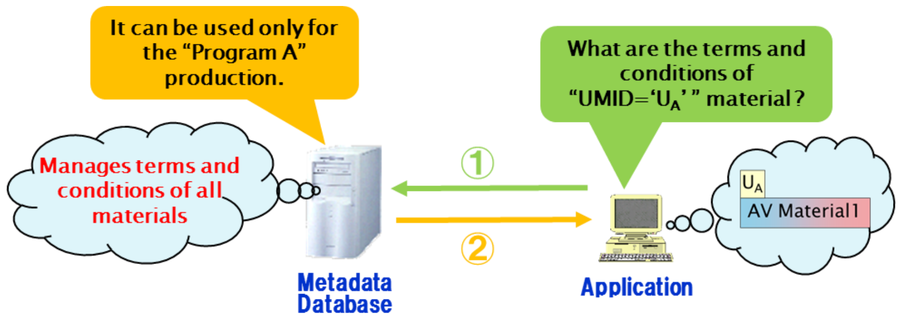

Figure C.9 – Additional Metadata Search

In this figure, an application ("Application") has a material uniquely identified by UMID 'U_A' at hand. A user of the application then wants to obtain the terms and conditions of the material and thus will give a query to the metadata database ("Metadata Database") that manages such information as metadata accordingly. The metadata database will then reply to the application with the requested information so that the application can display it to the user.

In this application scenario, though the metadata database behaves in the opposite way to that discussed in Annex C.4, it ought to be no big burden for the metadata database because it eventually manages the relationship between a UMID and metadata, which are associated via a material identified by the UMID.

It is worthwhile to mention that even if it is metadata associated with a material via a UMID, not the material itself, which is at hand in the application, the above scenario to search for the terms and conditions of the material can hold once the UMID value in question is extracted from the metadata, implying that the UMID value itself can work as bidirectional links.

Furthermore, even if a material at hand is the one derived from an original material in any way, the terms and conditions of the original material, which is often common or closely related to those for the derived ones, can be retrieved when the UMID attached to the derived materials are linking tools to the original one (see Annex C.2.1) because of the Material Number shared by all of those derived materials.

It ought to be noted that while some metadata might be preferably contained in a material (as a media file), there are other metadata that need to be managed separately from the material, In fact, which metadata ought to go with a material and which ought to be managed by a dedicated metadata database separately from the material is a fundamental design issue for a media production system, and therefore, additional metadata search by UMID as a key is also another important aspect of the UMID applications that ought to be paid attentions in the system design.

C.9 Loose Coupling between Application and Media Layers

As the market is rapidly changing such as with growing competition across a variety of delivery platforms including traditional broadcasting, mobile and the Internet, more attentions have been paid to the system that is flexibly configurable to meet a quick change of business requirements such as system scale-up or workflow changes. Thanks to the drastic increase of computational power and hardware resources available to hand, the SOA (Service Oriented Architecture) approach, which used to be only for the enterprise system in early days, has become popular even for the media production system, and relevant standardization activities such as SMPTE ST 2071-1 (Media Device Control) or FIMS (Framework for Interoperable Media Services) has taken place with much attention.

In such an approach as FIMS, each component that provides a certain function in a media production system such as ingest, transform and transfer is abstracted as a media service that is independent of a specific workflow, and with the help of so called SOA middleware, those services are dynamically integrated to form an actual media production system at a given workflow information.

Because each service is independent of a specific workflow, a system composed of those services is expected to be flexibly configured to meet a dynamic change of workflow. Furthermore, thanks to little or no interdependency among services, arbitrary services can be freely combined to form a system without unexpected side effects that are common for the reconfiguration of traditional media production system.

Such a characteristic of the SOA approach is called a loose coupling.

In addition, information exchanged among services is also abstracted as an XML instance in a SOA-based media production system, *i.e.*, even for the material treatment, it is an XML document describing a material as a proxy which is actually exchanged among media services until an actual material manipulation is certainly required, resulting in again contributing to a flexible integration of services.

If we apply this concept to the system shown in Figure C.5 , it is revealed that the system can be divided into two layers: the upper layer containing “Metadata Database” and “Application” in the figure and the lower layer containing various material servers as shown in Figure C.10 below.

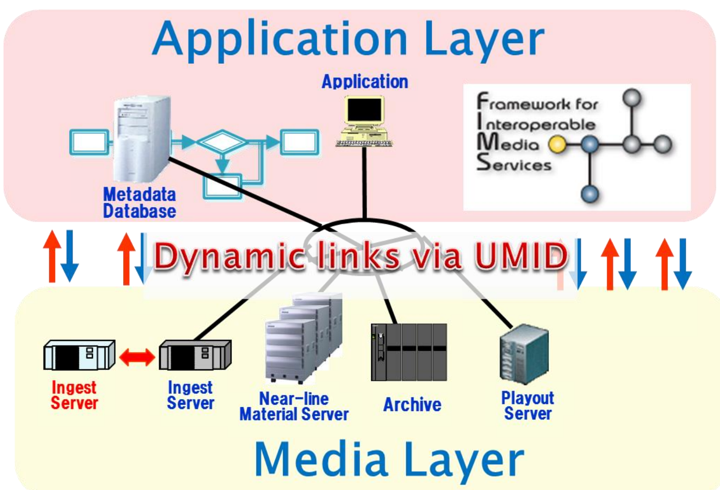

Figure C.10 – Loose Coupling between Application and Media Layers

In this figure, the upper layer, called the Application Layer, is the one where the abstracted media services are flexibly integrated to form a media production system for a given workflow. In this layer, although data that goes among services is usually tiny in its data size (typically as an XML instance), it needs to be handled in a very flexible and complex way.

The lower layer, called the Media Layer, is the one where materials are actually manipulated. Although material as data is usually huge in its data size, the data treatment is simple and straightforward; *i.e.*, data intensive processing such as material creation, material movement or material transformation.

While the SOA approach has brought flexibility in a system reconfiguration, it is only within the Application Layer. In other words, even though we want to replace the ingest server ("Ingest Server") with its mirror in the event of failure or to increase the number of near-line material servers ("Near-line Material Server") in order to accommodate an increase of materials, they cannot be supported by the flexibility in the Application Layer because the workflow information itself remains unchanged.

Furthermore, the URL, which is typically used in a conventional system to associate the data in the Application Layer such as an XML instance describing a material with that in the Media Layer such as the material as a media file, further deteriorates the flexibility in such treatments of material servers because it requires an explicit representation of hostname that is usually tightly coupled with a particular material server.

The UMID, together with its resolution protocol, will address this issue. By introducing a UMID as a globally unique material identifier, the material related data in the Application Layer is logically associated with its target material in the Media Layer via the UMID. Then, the UMID is resolved to an access method for the target material when actual manipulation of the material is required, which is called the dynamic linking of the Application and the Media Layers via UMID.

This results in easy replacement of the ingest server with its mirror (containing the same materials as backup) in Figure C.10. Furthermore, because where to store a material in the Media Layer is abstracted as its UMID from the viewpoint of an application in the Application Layer, a change of the number of near-line material servers cannot be seen from the application in the figure. As a result, any specific physical treatment of the near-line material servers such as increasing the number of it for the load balancing can be conducted without influencing an application in the upper Application Layer at all.

Consequently, the UMID can be used to realize even the loose coupling between the Application and Media Layers in a SOA-based media production system, resulting in further enhancing the flexibility in the system.

C.10 UMID Applications in Traditional VTR/SDI Environment

C.10.1 Introduction

The UMID has been introduced into the industry since the year of 2000 when VTR connected via SDI (Serial Digital Interface) was still dominant in the media production system. For the final example of the UMID applications, it is worthwhile to briefly review how UMID has been treated in such a traditional environment.

C.10.2 What is material in tradition VTR/SDI environment?

According to the implication of UMID Application Principle 2 (UMID Creation), a bounded sequence of frames is attached with a UMID to uniquely identify the sequence as a material. Consequently, a sequence of frames recorded on a tape by the REC Start and Stop operations of VTR for example is regarded as a material with a newly created UMID uniquely identifying the sequence. The REC Start and Stop operations that follow then create a new material which is recorded immediately after the previous one on the tape, and so on.

In general, because only a part of material such as a frame is accessible at a certain point of time for the linear recording device such as VTR, every frame in a sequence has the UMID which uniquely identifies the sequence as a whole, and each frame is recorded together with the UMID on a tape, enabling the UMID being accessible at any part of the material on the tape.

Note: There is an existing practice that, with a special tape cassette having an IC memory chip attached to it, a UMID uniquely identifying a material on its tape is recorded in the memory chip together with the In and Out points specifying the material on the tape. Furthermore, there is an experimental trial where an external application is used to separately manage a UMID uniquely identifying a material on a tape together with the tape identifier and the In and Out points to define the material on the tape, which is uniquely identified by the UMID.

It ought to be noted that because it is VTR which creates a new UMID and attaches it to the material being recorded on a tape, there is no collision of UMIDs even when a single source signal is simultaneously recorded on more than one VTR, which is in fact applicable even to the modern file-based media devices such as the ingest server.

For the VTR playback, on the other hand, a UMID attached to a frame on a tape is embedded as is into the SMPTE ST 291-1 ancillary data packet of the frame. Figure C.11 below schematically illustrates how a series of the materials as bounded sequences of frames originally recorded on a tape (“Material U_1”, “Material U_2” and “Material U_3”) with their respective UMIDs (newly created UMIDs of new Material Number (Mat.#) and zero Instance Number (Inst.#)) are transferred from the Source VTR to the Receiver VTR via SDI.

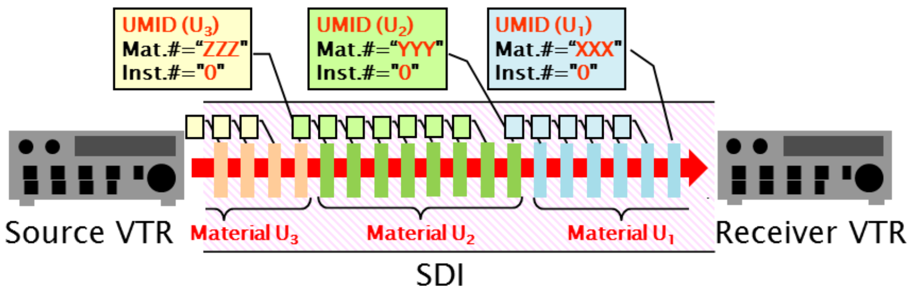

Figure C.11 – Materials with UMID over SDI

C.10.3 Two functions of UMID in traditional VTR/SDI environment

As is discussed in Annex C.2.1, there are two types of UMID to be attached to the materials for their recording at the Receiver VTR as shown in Figure C.12 and Figure C.13 below.

Figure C.12 below demonstrates the cases where materials are recorded at the Receiver VTR as the original materials with newly created UMIDs of new Mat.# and zero Inst.# being attached, which corresponds to the UMID as a globally unique material identifier shown in Figure C.1. Because how to define a new material as a bounded sequence of frames is left up to the Receiver VTR, it might record the materials while preserving the original durations of the respective source materials (Material U_4 to Material U_6), might record the materials with completely different durations (Material U_7 and Material U_8), or even might record as a new single material (Material U_{10}).

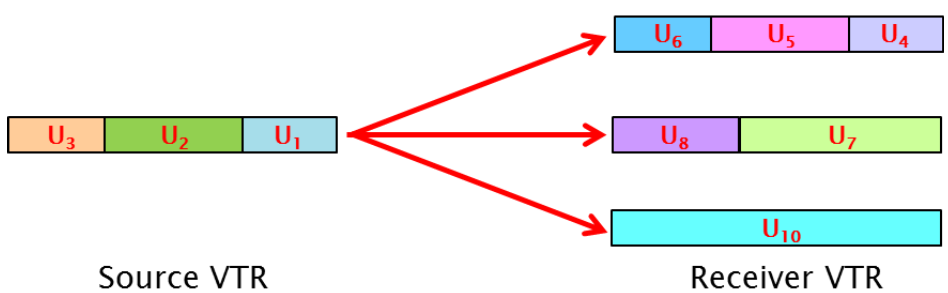

Figure C.12 – UMID to be Attached for Original Materials

It needs to be noted that, in general, even if the duration of a source material is kept for the recorded material, the UMID needs to be replaced with a newly created value in this case. This is because of the characteristic of VTR which generally adopts a lossy compressed material recording, *i.e.*, because the SDI requires for the material at the Source VTR to be decoded to the baseband signal, which is then recorded in a lossy compressed fashion again at the Receiver VTR, the materials in the Source and Receiver VTRs can no more be identical even if they share the same duration according to the UMID Application Principle 4 (UMID Identification).

Figure C.13 below, on the other hand, demonstrates the case where materials are recorded at the Receiver VTR as those derived from the source materials with UMIDs linking to respective source materials; *i.e.*, the UMID as linking tools shown in Figure C.2. In this case, a UMID of Mat.# being inherited from that of the source material and non-zero Inst.# is attached to the material that is recorded at the Receiver VTR.

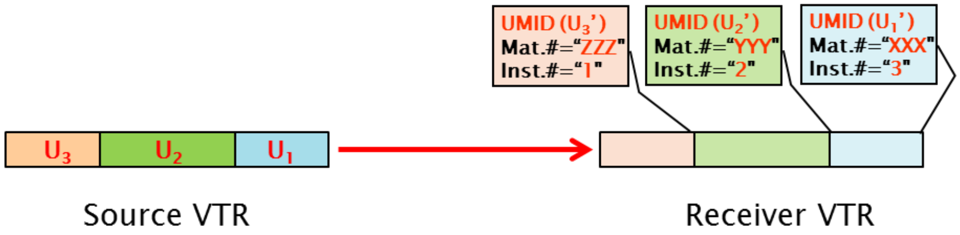

Figure C.13 – UMID to be Attached for Derived Materials

Because it is the UMID attached to each frame which is to be updated with the inherited Mat.# and non-zero Inst.#, the duration of a source material is usually preserved for the recorded material in most implementations. But there is no reason to exclude an implementation of the Receiver VTR that intentionally changes the boundary between adjacent materials on a tape, resulting in different durations of the recorded materials. This is justified by the UMID as a linking tool which is no more than just associating the material being recorded with its source material irrespective of their durations.

Note: If an end point of material U_1 (the left edge of the U_1 box at the Source VTR in Figure C.13) is extended over the material U_2 at its recording as the U_1' box at the Receiver VTR in the figure, it will make sense for the frames at the extended part of the material U_1' to have also the UMID U_1' rather than U_2' so that the UMID attached to any frame in the recorded material can uniquely identify the material U_1' as a whole. If the UMID function as a linking tool back to the source material is more emphasized, however, the frames at the extended part of the material U_1' ought to have had the UMID U_2' even though an intention is to record the material that is uniquely identified by U_1'. This also exposes the fact that the UMID functions as a unique material identifier and as a linking tool cannot be compatible but exclusive.

It is worth noting that, for the UMID as a linking tool, a useful option for the traditional VTR/SDI environment is provided by SMPTE ST 330, which is the copy number management. As is mentioned above, because of a lossy compressed material recording adopted by VTR, the repetition of material copy operation, or also known as the tape dubbing via SDI, causes a serious degradation of a material in its quality due to the iteration of baseband decoding and lossy compression. It is therefore crucial to know the copy generation number of a material at hand.

According to SMPTE ST 330, one byte copy number immediately followed by two bytes PRS (Pseudo-Random Sequence) generated number can be allocated to the three byte Inst.# field when the value of 3_h is specified at the Instance Number Generation Method Field or the 4 least significant bits (LSBs) of byte 12th of the UMID UL.

When it is specified, the copy number in the Inst.# field is incremented by one at every copy operation of a material. Because the Inst.# field of a newly created UMID for an original material is zero-filled, this copy number indicates how many copy operations have been applied to a material since it is originally created.

Note: Strictly speaking, the copy number in the Inst.# field does not always indicates a true copy generation number of a material since it is initially created because a newly created UMID with zero Inst.# is attached to a material whenever it is *deemed* as the original, or at its entry to the UMID based material management, regardless of its true copy generation number. In other words, it is the number of copy operations applied to the material since it is regarded as the original (See the item (d) in Figure C.3 whose copy number will be three for example). Hence a care ought to be taken to define the entry point of a material to the UMID based material management, or the point where a material is deemed as the original, in the design of the media production workflow chain.

It is also worthwhile to note that because UMID is attached to every frame for the traditional environment, the copy generation number is also managed per frame. As a result, the UMID attached to any frame in a composite material can convey the number of copy operations applied to a particular frame since its initial creation when the material composition workflow is appropriately designed.

C.10.4 Extended UMID and Source Pack

Another item that has been intensively employed in the traditional VTR/SDI environment is the Extended UMID. As is introduced in Annex A.2, the Extended UMID is provided to uniquely identify a finer granularity of a material called a material unit, which is composed of the 32-byte Basic UMID immediately followed by the 32-byte Source Pack describing “When”, “Where” and “Who” originally creates a particular material unit such as a frame. Consequently, a bounded sequence of frames as a material has a double layered identification scheme by the Extended UMID as schematically shown in Figure C.14 below.

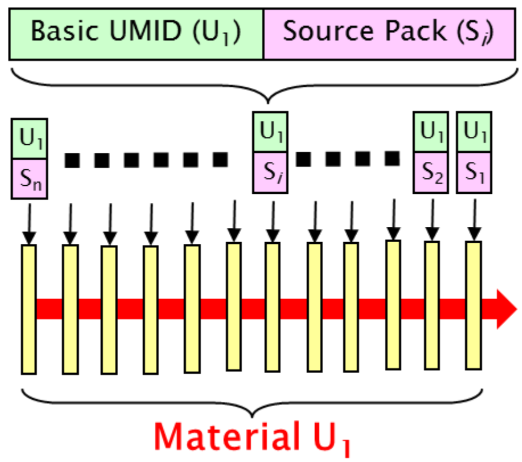

Figure C.14 – Material Identified by Extended UMID

According to this figure, it is obvious that the Source Pack for each frame needs to be distinct, which is usually achieved by its “When” component having higher resolution than the reciprocal of the frame rate. Furthermore, because a material as a whole is to be uniquely identified by the basic part of Extended UMID attached to any frame contained in the material, all the Extended UMIDs attached to the frames must share a single UMID value for its basic part, which constitutes the UMID Application Principle 6 (Extended UMID).

C.10.5 Source Pack applications

The Source Pack in the Extended UMID is a compact placeholder to accommodate and to deliver the fundamental metadata that “When”, “Where” and “Who” originally creates a material at the granularity of its material unit such as a frame, resulting in the Source Pack value varying along with the frames.

Because of its properties, the Source Pack can be used for various attractive applications, among which a couple of examples are introduced below.

One example of the Source Pack use is for the material browsing. Because the Source Pack that synchronizes with frame is delivered to the material browsing tool via SDI, it can be also “played back” with a frame if the browsing tool has a capability to detect and to represent the Source Pack on the display as appropriate. Figure C.15 below demonstrates such a Source Pack application.

Figure C.15 – Material Browsing with Source Pack

In this figure, a material that captures a seaside scene at typhoon is displayed with the time and place information for its acquisition being superimposed. Because the Source Pack varies with frames, the superimposed information is also continuously updated during the material playback as if it is also "played back" together with a frame.

Another example of the Source Pack use is for the material search. Because metadata accommodated by the Source Pack is one of the most fundamental informational items to characterize a material, it is effectively used to search for a desired material. Figure C.16 below demonstrates such a Source Pack application.

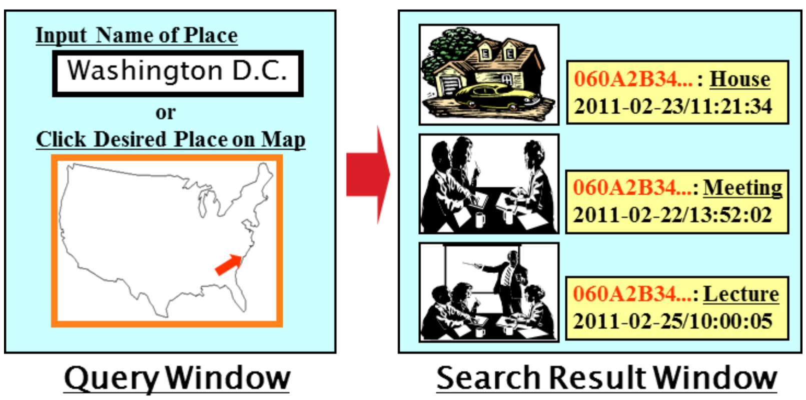

Figure C.16 – Extended UMID based Material Search

In this figure, a query and search result windows are shown side-by-side. A query for desired materials is given either by entering the name of a place as text or by pointing to the place on a map in the query window. Then, the resulting candidate materials appear in the search result window as their thumbnail pictures together with their UMIDs, their creation date/times, and their keywords if provided (how to accommodate thumbnail pictures and textual metadata such as keywords is beyond a scope of UMID application, though).

Note that behind such an application exists a system like the UMID based material search discussed in Annex C.4, and it is implicitly assumed that all the Source Packs originally embedded on materials are extracted, collected together with other metadata such as their thumbnail pictures and keywords, and stored into the metadata database as a prior registration of the materials into the material search system.

Finally, a Source Pack application that comes from another original intention for its introduction into UMID is introduced.

The original intention for the Source Pack is that it is expected to be used to track a material that travels through the production workflow chain so that the audit trail can be maintained. Therefore, it has been strongly encouraged to keep the Source Pack as is since it is initially created regardless of the treatment of the basic part of Extended UMID. In other words, even though the basic part of Extended UMID is replaced with a newly created UMID value with zero Inst.# for the new entry of a material as the original one, resulting in losing the association of the material with its source, the Source Pack is still encouraged to inherit as is, enabling to preserve a certain level of association between them.

As a result, the Source Pack is also expected to provide limited but useful information for the disaster recovery; *e.g.*, to recover from a failure to break the association between metadata database and material servers via UMID. Furthermore, because the Source Pack is associated with a material unit such as a frame, the recovery can be conducted even only a part of material in question is available to hand.

It ought to be noted, however, that because the assignment of a newly created UMID with zero Inst.# means the new entry of a material as the original one, a recording device has discretion to intentionally replace the Source Pack with the value that reflects the new entry. Furthermore, for some security reason, the recording device might delete the Source Pack from a material at all at its recording, which is also an acceptable treatment of the Source Pack.

In view of the original intention, on the other hand, it does make sense that the existing Source Pack cannot be replaced with a newly created value if the basic part of Extended UMID is just updated with the inherited Mat.# and non-zero Inst.# because this indicates that the material to hand is a derived one from a certain source material, not the original one. This constitutes the UMID Application Principle 7 (Source Pack), though the deletion of Source Pack still remains as an option to be taken even for this case.

C.10.6 Limitation of UMID applications for traditional VTR/SDI environment

While the primary function of UMID is as a globally unique material identifier, there is an obvious limitation for such a UMID function in the traditional environment because of typical implementations of UMID in a linear recording device such as VTR. As is mentioned in Annex C.10.2, a UMID uniquely identifying a bounded sequence of frames as a material is attached to every frame within the material in order for the UMID to be accessible at any part of the material on a tape. This, however, causes inconsistency when a part of material is overwritten with anther material by, *e.g*., an insert edit as schematically illustrated in Figure C.17 below.

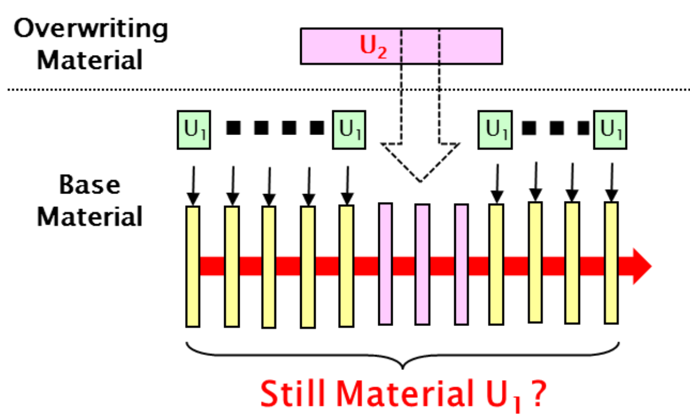

Figure C.17 – Material after Insert Edit

In this figure, three frames around the center of a base material are overwritten with frames from another overwriting material. Because a resulting material as a whole differs from the original base material at its overwritten part, the UMID attached to the base material must be replaced with a newly created UMID of zero Inst.#, say U_3, according to the UMID Application Principle 3 (UMID Integrity). Theoretically, it is realized by replacing all the UMIDs attached to the frames in the resulting material with a newly created UMID value (U_3). In reality, however, it is unfeasible because it is only the overwritten part in which the UMID can be replaced during a practical insert edit operation.

Note: Although the UMID Application Principle 3 (UMID Integrity) seems met if the UMIDs of overwritten frames remain as their original (‘U_1’) and nonexistence of an identical material with the original base material elsewhere is logically proven, the UMID Application Principle 4 (UMID Identification) cannot be met because of different playout result between the original base material and the resulted one. See Annex B.2.6 for the related discussion.

Furthermore, because a traditional linear recording device such as VTR usually does not have such a capability as to appropriately manage the UMID as a globally unique material identifier, an external application that compensates for the UMID managing function and is always running in conjunction with the recording device is required.

Therefore, with the insert edit as a common practice for the traditional VTR/SDI environment, the UMID as a globally unique material identifier cannot but be regarded as impractical.

As for another UMID function, or the UMID as a linking tool, on the other hand, the consistent handling of UMID as a unique material identifier is not necessary; *i.e.*, with a UMID of the inherited Mat.# and non-zero Inst.#, what is provided by the UMID is just an association between the source and resulting materials. Furthermore, together with the benefit of the copy number management discussed in Annex C.10.3, one can conclude that the UMID as a linking tool is more practical and useful for the traditional VTR/SDI environment than the UMID as a globally unique material identifier as has been observed in the existing practices.

Note: Regarding the UMID as a linking tool, UMIDs attached to the frames to be overwritten with an overwriting material (U_1 in Figure C.17) also need to be replaced with those of the overwriting material (U_2’ where “apostrophe” denotes the UMID of inherited Mat.# and non-zero Inst.#), resulting in that any individual frame in the resulting material is to be linked back to the source material it originally comes from.

It needs be also noted that the Source Pack treatment introduced in Annex C.10.4 is basically independent of the Basic (part of Extended) UMID functions. Furthermore, taking account of the Extended UMID being attached to every frame in a material for the linear recording device such as VTR as discussed in Annex C.10.2, the Source Pack, which describes “When”, “Where” and “Who” originally creates *a frame* rather than the material as a whole, can be treated intuitively on its own for various UMID applications. Therefore, also by taking its functionality as tracking back to the source material into consideration, it is natural to observe that the Source Pack has been much more exploited than the Basic (part of Extended) UMID for the existing practices in the traditional VTR/SDI environment.

Annex D Frequently Asked Questions (Informative)

Q 1: What is the UMID and what problem does this Recommended Practice address?

The UMID (Unique Material IDentifier) is a unique audiovisual material identifier standardized by SMPTE as SMPTE ST 330 and SMPTE RP 205 in 2000. Although UMID has been widely disseminated over the industry as a mandatory component for the MXF (Material eXchange Format) technology since its introduction, the originally intended use of UMID as a globally unique material identifier to associate the material with its external metadata had seldom been seen in practice for a long time. This is partly because of lack of standardization of golden rules applicable to all kinds of UMID applications, which must be strictly followed by all UMID-aware audiovisual products in order to realize the UMID applications over the products from multiple vendors, which is called the UMID Application Principles.

This Recommended Practice normatively specifies the UMID Application Principles and describes considerations on the UMID uses in its applications, together with informative text on the UMID basics, how to implement the UMID Application Principles for a UMID as a globally unique material identifier, various UMID application examples, and frequently asked questions on the UMID applications.

Q 2: What are the UMID Application Principles?

The UMID Application Principles are a set of golden rules applicable to all kinds of UMID applications, which must be strictly followed by all UMID-aware audiovisual products in order to realize the UMID applications over the products from multiple vendors. They are composed of seven items, or Principle 1 (Definitions), Principle 2 (UMID Creation), Principle 3 (UMID Integrity), Principle 4 (UMID Identification), Principle 5 (UMID Inheritance), Principle 6 (Extended UMID) and Principle 7 (Source Pack).

Q 3: What is the UMID Managed Domain?

The UMID Managed Domain is a conceptual domain composed of audiovisual materials with a valid UMID in the sense of the UMID Application Principles. This is in fact an embodiment of the UMID Application Principles. In order for the UMIDs attached to materials in the UMID Managed Domain to be always maintained valid in the sense of the UMID Application Principles, certain UMID treatments are usually required at every material manipulation in the domain.

Q 4: How does UMID differ from other existing identification schemes for audiovisual entity such as ISAN or IEDR?

Technically, the most particular characteristic of UMID from other identification schemes such as ISAN (International Standard Audiovisual Number) or EIDR (Entertainment IDentifier Registry) is that UMID can be created at any time without accessing the registration authority or the central database, which other identification schemes rely on to assign a globally unique value. Based on the UMID creation algorithm specified in SMPTE ST 330, one can obtain UMID of guaranteed globally unique value without negotiating any external entities.

This UMID characteristic is suitable to audiovisual material management in a media production system, where materials, either as the original or the derived ones, are created very frequently in the system. Note that, as for the materials in the media production, those having different resolutions or frame rates need to be strictly distinguished even though they share the same content. Furthermore, a truncated material resulted from a partial retrieval operation ought to be made distinct from its source material even though their difference is only one frame duration. As a result, UMID is, in general, to be created and attached to a material at every operational step in the media production workflow chain.

As for a program or content identification where it is often desired that the audiovisual entities sharing the same content are regarded as the same for some purpose irrespective of their technical properties such as resolutions or frame rates, however, UMID is not suitable even though technically it also could have been used for such a purpose.

Moreover, considering a situation where users who are unknown in advance often want to access the entities, an identification scheme with a registration authority is more appropriate because the registration needs to be made known in public for any future users. Furthermore, registration authorities often support alternate identifiers, so an entity identified with a UMID can be given a registered identifier and later looked up in the registry using the UMID. This accommodates use cases for both immediately allocated and publicly resolvable identifiers being demanded such as for the finished materials of various forms depending on distribution channels, each of which is uniquely identified by UMID (especially for it as an MXF file) while all of which as the same content are uniquely identified by the registered identifier.

In this way, the UMID as a globally unique material identifier and the registration authority based identification scheme (such as ISAN or EIDR) as a globally unique program or content identifier do not compete each other but can co-exist and to be used separately or together for their respective purposes.

Q 5: Why does UMID have to be globally unique?

Before answering this question, it ought to be noted that though UMID as a unique material identifier can be used without the registration authority, this does not mean that a central database kind is not necessary at all. In order for the materials in a certain local scope such as within a product or a subsystem to be made available to an external application, a material manager that recognizes and manages the materials existing in the scope is always required. It is the material manager's responsibility to recognize where to locate a material within the scope (which obviously requires making the material known to the manager when created) and to provide an external application with information to access the material in response to a request from the application.

The global uniqueness of UMID is effective when multiple scopes from local material managements are merged to obtain a wider material management scope. As long as the UMIDs are created strictly based on SMPTE ST 330, they all are guaranteed not to collide at all. Consequently, any local material managements (such as within a product or a subsystem) can be easily integrated to form a larger system without any prior adjustment, which is the biggest advantages of UMID as an industry standardized globally unique material identifier in a media production system.

Q 6: Do we need to replace our own house material ID with UMID when to introduce the UMID-aware audiovisual products?

The answer depends on what your house material ID specifically identifies or the operational rules of your house material ID. If your house material ID is regarded as equivalent with the UMID from its operational viewpoint, the UMID-aware product can behave as if it recognizes your house material ID (instead of UMID) by just introducing the house material ID to/from UMID mapping application.

If your house material ID is used to identify a program or content rather than the material, the UMID is orthogonal and can co-exist according to the discussion made in Q 4.

Ideally, as a common technological infrastructure for the Professional Media industry, it is more preferable for the UMID treatment to be invisible from a user application as schematically demonstrated in Figure D.1 below.

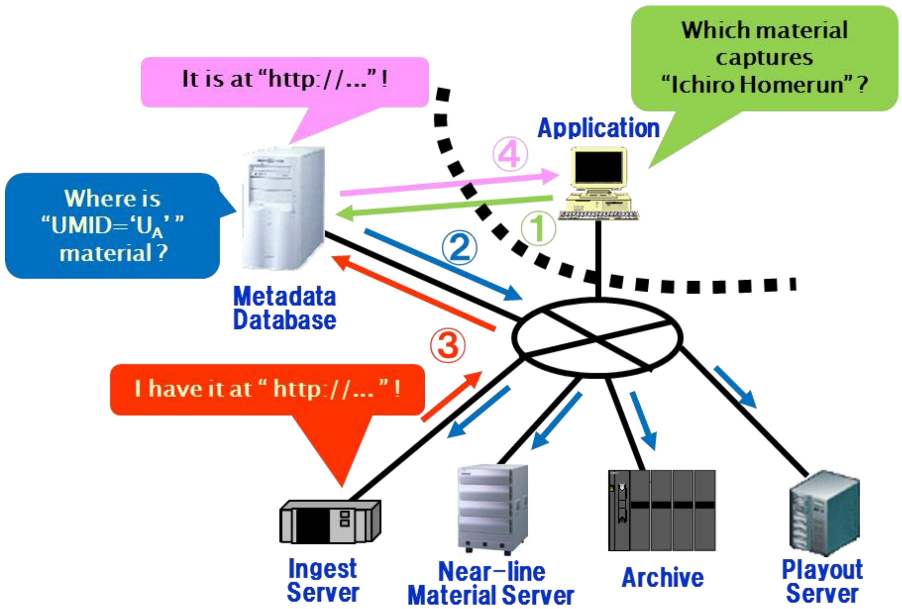

Figure D.1 – "Ideal" UMID based Material Search

Compared with the UMID based material search shown in Figure C.5, the UMID resolution in this figure is conducted between the metadata database ("Metadata Database") and the material servers ("Ingest Server", "Near-line Material Server", "Archive", and "Playout Server") instead of an external application ("Application") and the media servers in Figure C.5. Consequently, the UMID treatment is concealed from the application, resulting in for the application to access a desired material by just asking the initial query, without knowledge of UMID at all.

In such a system configuration, your house material ID can be used in the same way as before. Moreover, because certain functions your house material ID has provided so far are to be taken over by the UMID, the operations with your house material ID are to be minimized, or more focused on the functions specific to your own system requirements, though some harmonization between the UMID and your house material ID might need to be carefully designed in advance to achieve it.

It ought to be noted that the same UMID resolution protocol as used in Figure C.5 is adopted for the scenario shown in Figure D.1; *i.e.*, from the system implementation viewpoint, only the difference between the systems shown in those figures is where the UMID resolution protocol stack is implemented, *i.e.*, whether it is in the metadata database (for Figure D.1) or in the application (for Figure C.5), which needs to be determined based on the overall system design policy.

Annex E Bibliography (Informative)

Note: All references in this document to other SMPTE documents use the current numbering style (e.g. SMPTE ST 2029:2009) although, during a transitional phase, the document as published (printed or PDF) may bear an older designation (such as SMPTE 2029-2009). Documents with the same root number (e.g. 2029) and publication year (e.g. 2009) are functionally identical.

SMPTE ST 291-1:2010, Ancillary Data Packet and Space Formatting

SMPTE ST 2029:2009, Uniform Resource Names for SMPTE Resources

SMPTE ST 2071-1:2014, Media Device Control Framework (MDCF)

EBU Technical Review Special Supplement 1998, “EBU / SMPTE Task Force for Harmonized Standards for the Exchange of Programme Material as Bitstreams, Final Report: Analyses and Result”
http://tech.ebu.ch/docs/techreview/ebu-smpte-tf-bitstreams.pdf

Y. Shibata, et al., “Introduction to XDCAM Metadata”, Proc. NAB 2005 Broadcast Engineering Conference, pp.421-424 (2005)

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.

Framework for Interoperable Media Services (FIMS) available at: http://fims.tv/