SMPTE Object Identification

Prepared and presented by:
Jim Wilkinson, SMPTE Fellow
Sony BPRL, UK

---

SMPTE 298M (ULs)

- Titled: “Universal Labels for Unique Identification of Digital Data”
- Uses ISO/IEC object identifiers as specified by 8824-1
  - Provides a hierarchy through a sequence of identification nodes – from 4 bytes upwards in length in increments of 4 bytes
  - SMPTE lies directly under ISO as “ISO:SMPTE”
- SMPTE defines all its ULs as 16 bytes in length
  - 0x06.0E.2B.34. [+12 further bytes]
- ULs are not widely used outside of SMPTE
- Within SMPTE, ULs provide for interoperability of essence (pictures, sound, data essence) and all kinds of metadata
OIDs as Keys

- SMPTE Labels can be used both for data identification and for data coding.
- Coding is the well-known Key-Length-Value triplet used in many systems.
  - This is essentially a machine-level code.
  - Highly efficient coding with minimal overhead.
  - But not human-friendly.
- Coding is enshrined in SMPTE 336M.

SMPTE 336M (KLV Coding)

- Defines SMPTE Keys as ULs that use short-form BER coding.
  - Result: every byte is a separate codeword in the range 0x01~0x7F.
- Defines KLV coding for values that are categorised as:
  - Data Groups - multiple components of essence, metadata, control or other data where each component is a data element.
  - Data Elements - single components of essence, metadata, control and other data.
  - Data Types - identification and coding of information which defines the data representation of elements and groups.
- It also defines other forms of coding, notably:
  - Labels - ULs that are to be used as values alone and may be used as values in a K-L-V triplet.
SMPTE Registers

- SMPTE Registers define the usage of each SMPTE UL
- So there are registers for:
  - Groups (not much populated yet though)
  - Elements (well populated and well used)
  - Types (still in development)
  - Labels (well populated and well used)
  - And a prototype register for enumerated terms (textual and numeric enums)

- Each SMPTE is defined by:
  - A controlling standard that defines the structure of the register
  - The contents of the register (typically as Excel)

- All SMPTE registers are freely available on www.smpte-ra.org

Generalised Registry Model (WiP)

![Generalised Registry Model Diagram]

Notes on ‘Types’:
1. ‘Data Type’ is abstract and never directly instantiated but all concrete types are a sub-class of "Type".
2. A Base type is a basic unit of data representation that permits all values (e.g. UInt16).
3. A Constrained type is a base type that limits the range of data values (e.g. Chinese character values in UTF16)
4. An Enumerated Base type is a base type that has defined data values which may be numeric or textual.
5. A Constructed type may be composed of any combination of types of whatever form (base, constrained, enumerated or other constructed).
6. A Constrained Constructed type is a constructed type that has restrictions on the range of data values (e.g. limits on the range of values for an array).
7. An Enumerated Constructed type is a constructed type that has defined data values that may be either number or textual (note - strings are constructed types and can be enumerated as controlled terms)
Register Structures

The primary nodes of each register are defined as:

- Class 1: Identification and Location (local and global scopes)
- Class 2: Administration (business data including security)
- Class 3: Interpretation (how a value is to be interpreted - thesauruses)
- Class 4: Parametric (technical data for coding and compression)
- Class 5: Process (identification of processes and device or processor settings)
- Class 6: Relational (identification of how data relates to other data)
- Class 7: Spatio-Temporal (time data and geo-spatial coordinates)
- Class 12: Compound (not widely adopted)
- Class 13: Public Organisationally Registered
- Class 14: Private Organisation Registered
- Class 15: Experimental (a sandpit for testing - not interoperable)

SMPTE EG38 documents the full tree structure

Nodes and Leaves

- Nodes do not contain useful values but are used for collecting related register items in one place
- Leaves contain the items that define the register
Register Management

**SMPTE R30 Technical Committee**
- Governs the process by which the SMPTE Registration Authority manage, store and disseminate the information contained by individual registers.
- Currently the registers are authored in spreadsheet form
- Publication is currently in spreadsheet form but R30 is looking to publish the register in a web-friendly form for access by web browsers
- Work is focused on the automatic generation of web-based access through a publicly available (and free!) database solution

Public and Private Org Spaces

**Public Organisation space (class 13)** has been defined across all registers to allow organisations to register and publish their entries for open access
- This space is free, but requires the organisation to publish all its uses of their register space

**Private Organisation space (class 14)** requires a payment to the SMPTE-RA but does not require all entries to be published
- POA to the SMPTE-RA, currently $2500

In each case, the organisation has a space of 6 bytes allocation permitting a total of approx $4 \times 10^{12}$ values
**KLV Application Coding**

- KLV Coding of elements and groups can be concatenated to provide a ‘Scheme’
- Such a ‘scheme’ is MXF (Material eXchange Format)
- Now widely adopted throughout the professional content creation industries (both digital A/V and digital cinema)
- MXF is essentially built of the following KLV coded components:
  - KLV coded metadata
  - KLV coded audio-visual essence
  - KLV coded file indexing tools

---

**SMPTE UMID**

- UMID - Unique Material IDentifier (SMPTE 330M)
  - “a unique identifier for audio-visual material which is locally created and globally unique”
  - “Audio-visual material”:
    - Any one or any combination of picture (or video) essences, sound (or audio) essences and data (or auxiliary) essences. This term is also frequently referred to simply as “material”
- Basic and extended UMIDs
  - A basic UMID which contains the minimum components necessary for unique identification
  - An extended UMID which attaches a packed metadata group (aka ‘source pack’) to the basic UMID
UMID Data Structure

<table>
<thead>
<tr>
<th>Universal label</th>
<th>Ext No.</th>
<th>Material Number</th>
<th>Time/Date</th>
<th>Spatial Co-ordinates</th>
<th>Country</th>
<th>Org</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td>12 bytes</td>
<td>3 bytes</td>
<td>16 bytes</td>
<td>8 bytes</td>
<td>12 bytes</td>
<td>4 bytes</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

Extended UMID (64 bytes)

Source Pack (32 bytes)

XML Transformations

- KLV is wonderful for machines
  - Very fast, low overhead, efficient, +++
- But not so good for humans...
  - Humans do not read ULs in the same way as textual words (well, most humans!)
- To serve the need for exchange of KLV metadata and the world of XML, the registers are adding symbol names specifically to provide a unique name for each entry
  - Schemas are required to make sense of the relationship between KLV coded items at the scheme level
  - Work progresses...
Summary

- All SMPTE work is now based on 16-byte Universal Labels for both identification and coding of data.
- This approach permits a consistency of coding for all data constructs - whether the data be simple elements, groups of elements or other data kinds.
- ULs are machine efficient but human antithetic
  - Mappings to human readable (XML) form now in progress
- All SMPTE register work is freely published by the registration authority at: www.smpte-ra.org
  - Currently the metadata and labels registers are listed