THE QUEEN'S AWARDS FOR ENTERPRISE 2001

BBC

R&D White Paper

WHP 076

September 2003

Enabling interoperable IT-based programme-making tools with AAF & MXF

S. H. Cunningham ***and*** **P. N. Tudor**

Research & Development
BRITISH BROADCASTING CORPORATION

BBC Research & Development
White Paper WHP 076

Enabling interoperable IT-based programme-making tools with AAF & MXF

S. H. Cunningham and P. N. Tudor

Abstract

Many organisations are contemplating the move to new programme-making tools based on generic IT equipment. There is remarkable consensus among key manufacturers and end-users that the Advanced Authoring Format (AAF) and related Material eXchange Format (MXF) will be the key standards for the interchange of programme material within IT structures.

This paper addresses the scope of application of AAF and MXF. Firstly, the underlying data model is examined and related to concepts familiar to programme makers. Secondly, the use of different essence codecs is described including MPEG-2 long GOP. Thirdly, the application of MXF as a storage format for editing systems is discussed. Finally, the status of implementations of AAF, MXF and MPEG-2 in open-source applications and development libraries is reviewed.

This document was originally published in the Conference Publication of the International Broadcasting Convention (IBC 2003) Amsterdam 11th-15th September 2003

White Papers are distributed freely on request.

Authorisation of the Chief Scientist is required for publication.

ENABLING INTEROPERABLE IT-BASED PROGRAMME-MAKING TOOLS WITH AAF & MXF

S. H. Cunningham and P. N. Tudor

BBC R&D, UK

ABSTRACT

Many organisations are contemplating the move to new programme-making tools based on generic IT equipment. There is remarkable consensus among key manufacturers and end-users that the Advanced Authoring Format (AAF) and related Material eXchange Format (MXF) will be the key standards for the interchange of programme material within IT structures.

This paper addresses the scope of application of AAF and MXF. Firstly, the underlying data model is examined and related to concepts familiar to programme makers. Secondly, the use of different essence codecs is described including MPEG-2 long GOP. Thirdly, the application of MXF as a storage format for editing systems is discussed. Finally, the status of implementations of AAF, MXF and MPEG-2 in open-source applications and development libraries is reviewed.

INTRODUCTION

The performance of generic IT equipment is increasing on all fronts – processor speeds, storage volumes and network capacities. This is enabling a change in the nature of programme-making tools, away from stand-alone systems with traditional tape-based input and output, towards a more distributed collaboration of tools based on networked IT platforms, sharing material and metadata (1).

However, to realise this change with tools from multiple manufacturers, agreement must be reached on interoperability standards. One area requiring standardisation is the file format for interchanging material and associated metadata. There is now remarkable consensus that the Advanced Authoring Format (AAF) and Material eXchange Format (MXF) will be key formats for successful cross-platform, cross-manufacturer file interchange.

Both formats draw on a common metadata structure, registered in the SMPTE metadata dictionaries (2). These define individual metadata items, sets of metadata items, types and controlled values. MXF and AAF encode and transport these data structures in specific ways, optimised to their fields of application.

The support for AAF & MXF is broad. Several large programme-making users seeking to deploy key standards throughout their organisations have stated their commitment to basing future purchasing decisions on support for these formats. Many equipment manufacturers also recognise that the growth opportunities that standardisation of the interchange format bring outweigh the perceived risks of “opening up” their platforms.

AREAS OF APPLICATION

MXF is a file format for source material and finished programmes. It was developed by the Pro-MPEG Forum (3) and AAF Association (4), and is standardised as SMPTE 377 – 394M.

It is optimised for sequential reading & writing, and is suitable for implementation in a wide range of devices, from cameras & VTRs to PC-class devices. It is a container and wrapper for material, i.e. many types of essence (compressed or uncompressed) can be carried within it, along with metadata describing that essence.

AAF is a superset of MXF and is suitable for source material and sophisticated editing metadata. It was developed by the AAF Association who provide a software reference implementation (5). It is optimised for editing applications and is predominantly aimed at PC-class devices.

Whilst MXF and AAF are effective on their own, they have been designed to work well together, as shown in Figure 1. In this example, the ingest process creates MXF material files which are written to shared storage. These files may be accessed directly by an off-line editor to create a rough-cut. The edit metadata describing the rough-cut is transferred to a craft editor for finishing; the craft editor follows the references in the AAF file to the MXF material. Finally, the "rendered" finished programme is produced by the craft editor and written back to the storage as MXF, for transfer to a delivery process.

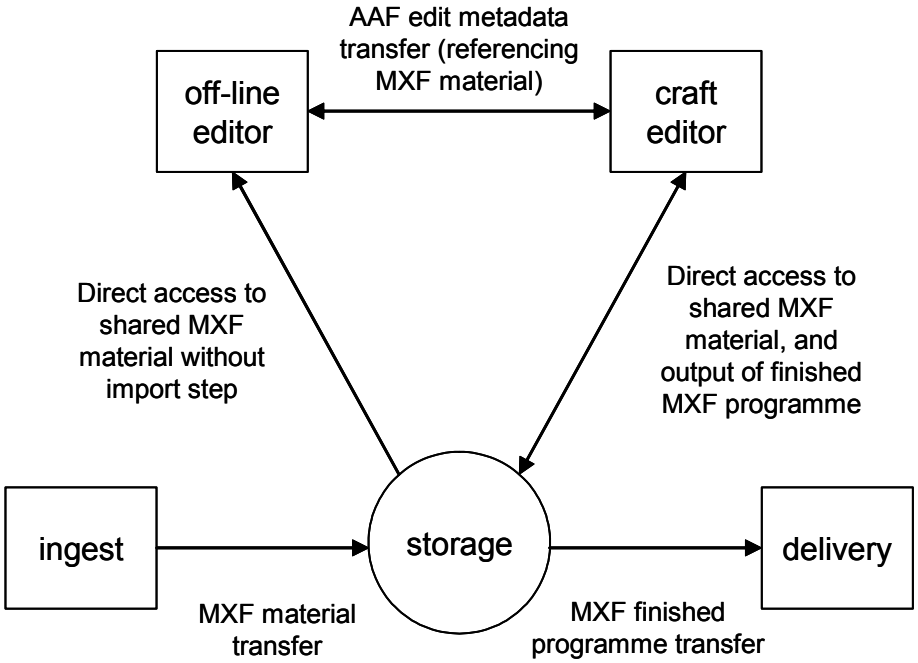

Figure 1 – Scenario using MXF material with AAF edit metadata

METADATA STRUCTURE

The metadata within an AAF or MXF file is structured according to a data model. This defines the logical contents of the file: what the data structures are and the associations between them. The AAF data model describes compositions (i.e. creative edit decisions), digitised material and physical source tapes and films. MXF re-uses a subset of these structures to describe digitised material.

The AAF data model is object-oriented, which gives the following advantages:

- Objects provide a framework for containing and labelling different kinds of information.
- Objects make it possible to treat different items in the same way for attributes they share. For example, with an AAF or MXF file one can find out the duration of video data, audio data, timecode data or descriptive data, without having to deal with their differences.

Similarly, one can play audio or video data either contained within an object, or stored in an external file and referenced by an object.

- When the information becomes very complex, objects provide a mechanism to describe it in a structured way.

A central class of object in the AAF (and hence MXF) data model is the “Package” (this is called “Mob” (Metadata Object) within AAF software tools). A Package holds the metadata description of all kinds of essence; be they physical tapes or films, digitised material, or compositions.

Each Package has a globally unique identifier for the piece of essence and its metadata. The value used for this 32-byte unique identifier is a SMPTE Unique Material Identifier (UMID) (6).

The metadata held in the Package includes a name, date & time, an overall description of the essence (e.g. file, tape or film details and possibly location) and a description of the tracks. The track metadata describes the synchronisation relationships between the tracks and also the components of essence within each track. This is shown in Figure 2.

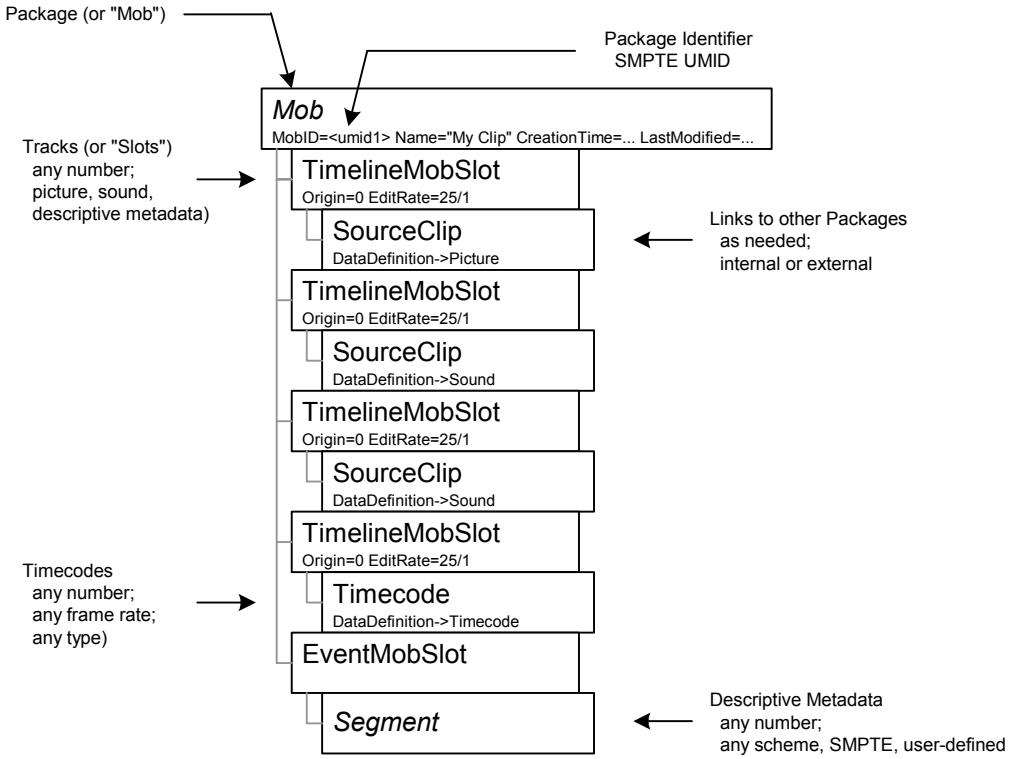

Figure 2 – A typical AAF or MXF Package object

Programme making is intrinsically about manipulating and assembling source material into new (derived) pieces. These in turn may be manipulated and assembled in some fashion, leading to a “chain” of manipulation and assembly to get from the first source materials to the final composition. The AAF metadata describing this process is structured in an identical manner.

Each Package describes a step in the chain. The metadata within each track specifies how the previous step in the chain has been used.

A typical example of a Package chain is shown in Figure 3. Here, we describe (from right to left) how an original source on film was scanned by a telecine to produce a video tape. Next, we describe what portion of the video tape was digitised into an editing tool to create a file.

Then, we describe how the digitised file relates to what the editing tool portrays as the clip. Finally, the creative edit decisions of the composition are described in terms of clips.

The point to note is that the AAF data model does not “flatten” this metadata into a single big list of attributes, but rather creates a separate description of each step in the chain, and links these together. This is a powerful approach that scales to more complex scenarios.

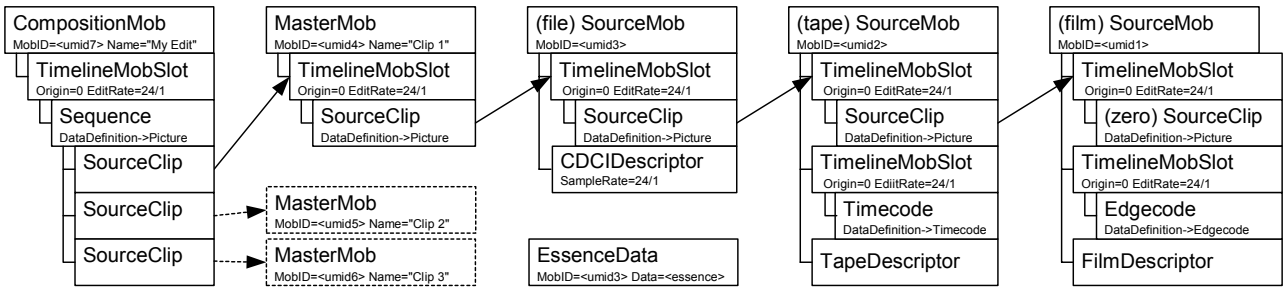

Figure 3 – A typical Package chain

ESSENCE CODECS

Describing the essence

In a networked IT environment, systems will encounter a variety of compression types as different parts of the business, having made different equipment choices, seek to join up. Software codecs allow flexibility for an IT platform to handle multiple compression types.

A feature of MXF and AAF is their unambiguous standardised source metadata such as picture size, frame rate and compression type. Figure 4 shows the AAF metadata associated with a video file Source Package (Mob). Such standardised metadata provides the foundation for supporting multiple essence codecs.

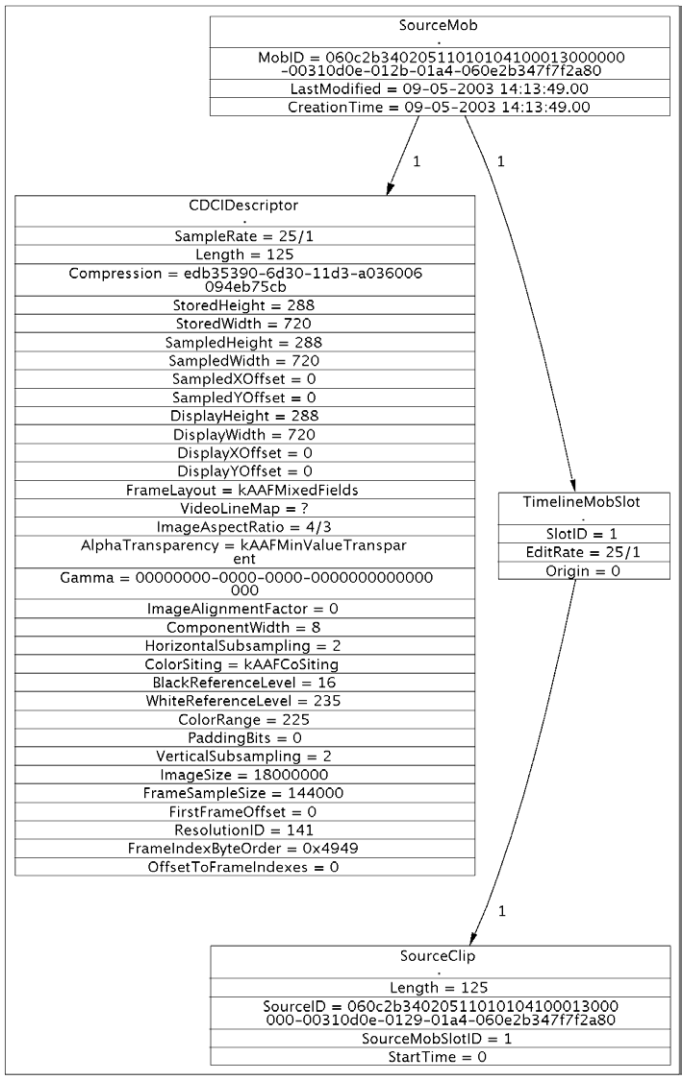

Figure 4 — A video file Source Package

AAF plug-in codecs

Plug-in codecs are supported in several media sub-systems such as Apple Quicktime (7) and Microsoft Windows Media (8). In such sub-systems, an interface is defined to the codec. It is typically a frame-based interface even when a compression scheme uses inter-frame coding.

The AAF reference implementation (5) establishes a frame-based plug-in codec Application Programming Interface (API). This should exist in any AAF application based on the AAF reference implementation. By creating a codec that implements the defined API, and

plugging it in to an AAF system, support for additional compression types can be added.

The MPEG-2 long-GOP codec is an attractive essence format for its low storage costs and existing acceptance in digital television broadcasting. The issues of using a codec exploiting inter-frame redundancy in a video editing system were studied by Brightwell et al (9). A promising way of exploiting this work is to create an AAF MPEG-2 long-GOP codec which would provide a frame-based API to the video editing application, isolating the complexity of the underlying inter-frame based codec.

MXF AS AN EDITING STORED FORMAT

MXF and AAF are primarily intended to be interchange formats – two systems may use these formats to interchange material even though neither may use them internally to store the material. In this case, the systems would use import and export functions to copy between the interchange format and the internal format.

However, there are scenarios where direct working with MXF or AAF files is attractive, without an intermediate copy step. One particular scenario is shown in Figure 1, where two editing systems, potentially from different manufacturers, are working directly with a single stored MXF copy of the material on shared storage. In this case, copying the material to the particular internal format used by each editor is time consuming and wastes storage capacity. A version of MXF that is suitable for this scenario is being developed and documented as a SMPTE standard. An MXF Operational Pattern (OP) is the formal way in which constrained versions of MXF are standardised.

The MXF OP for use as an editing stored format is called OP–Atom. Its key features are summarised below.

- OP-Atom files have a simple, predictable layout with minimum scope for variation, to allow efficient real-time or faster than real-time access.
- OP-Atom files always store the essence with its metadata in the same file (unconstrained MXF allows the metadata to be stored in a separate file to the essence).
- OP-Atom files store each essence track in a separate file. This provides efficient access to the essence for editing applications which are typically only interested in a single track from each essence source. The metadata in each file describes the relationships between the multiple tracks as originally digitised.

STATUS OF IMPLEMENTATIONS

In any new file standard, ready availability of reference implementations to encourage development is important. For example, the MPEG Software Simulation Group (MSSG) (10) released an MPEG-2 video codec reference implementation in source code form providing the ability to study, use, modify and derive improved forms of the software. Therefore, open source implementations are particularly relevant to encourage proliferation of AAF and MXF.

We now review the status of implementations of AAF, MXF and MPEG-2 in open source applications and development libraries.

AAF open source software

AAF reference implementation (AAF Association)

The reference implementation of the AAF Specification is distributed as an open source Software Development Kit (SDK). The SDK is platform independent and provides an object

oriented API in C++. Since the AAF Engineering Committee oversees development of the SDK, a high degree of agreement between the Specification and the reference implementation is maintained. The SDK can be used directly by manufacturers to add AAF support to their products.

A number of features important to developers are included in the SDK in addition to the reference implementation. A regression test suite is provided to test correct functionality of the toolkit. Example client applications illustrate how to use the API. Extensive documentation is provided including a developers' guide, an API reference and the AAF Specification.

aafviewer (BBC)

The aafviewer application displays a graphical representation of an AAF file showing the AAF objects and the relationships between them. The user can navigate around the AAF instance using pan and zoom mouse controls. References between objects are highlighted in colour. Figures 4 and 5 show two different views of the same AAF file. aafviewer has been recently contributed to the AAF SDK source distribution.

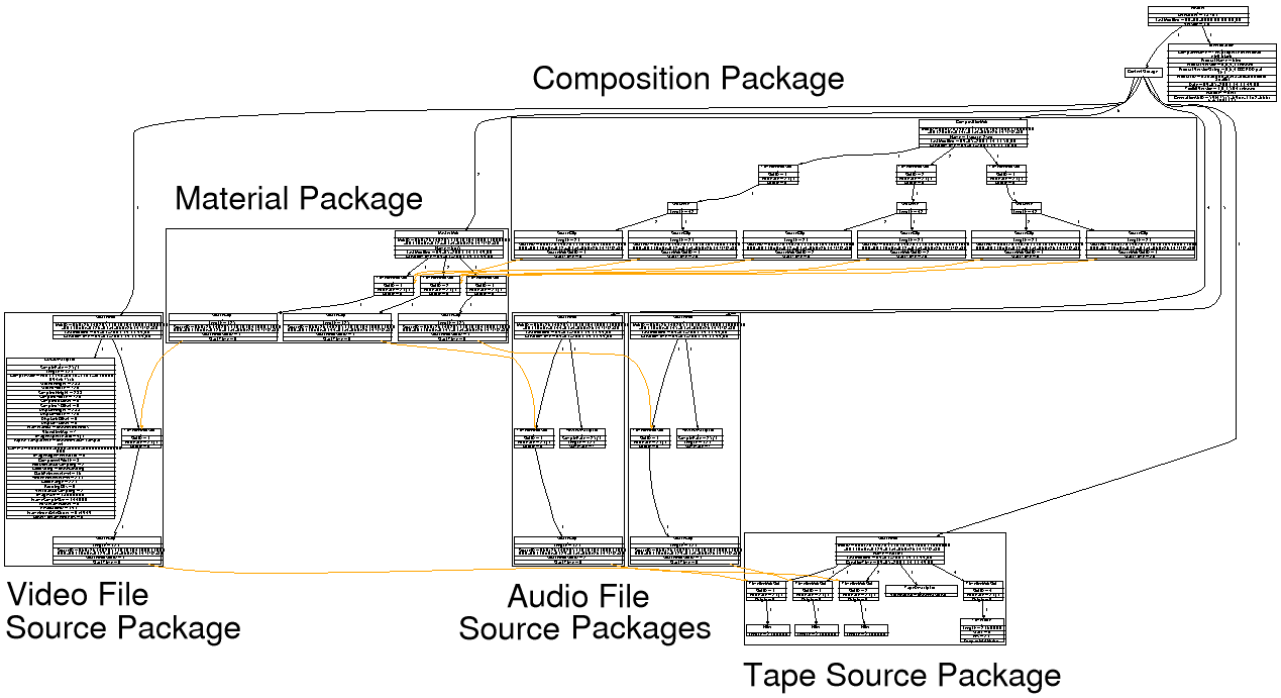

Figure 5 – aafviewer displaying an AAF file (with additional text annotations). An exploded view of the video file Source Package is shown in Figure 4.

AAF support in Kino (BBC)

Kino is a non-linear DV editor for the GNU/Linux platform. It supports DV ingest from an IEEE 1394 device such as a DV camcorder, creation of cuts-only compositions and export to AAF with embedded DV essence. These features were demonstrated at NAB 2003 & IBC 2003 where live footage of attendees was captured and edited in Kino, exported as an AAF file and finally imported into Avid Xpress DV (see Figure 6). The AAF-export functionally of Kino has been separated into a stand-alone application named eli2aaf and recently contributed to the AAF SDK source distribution.

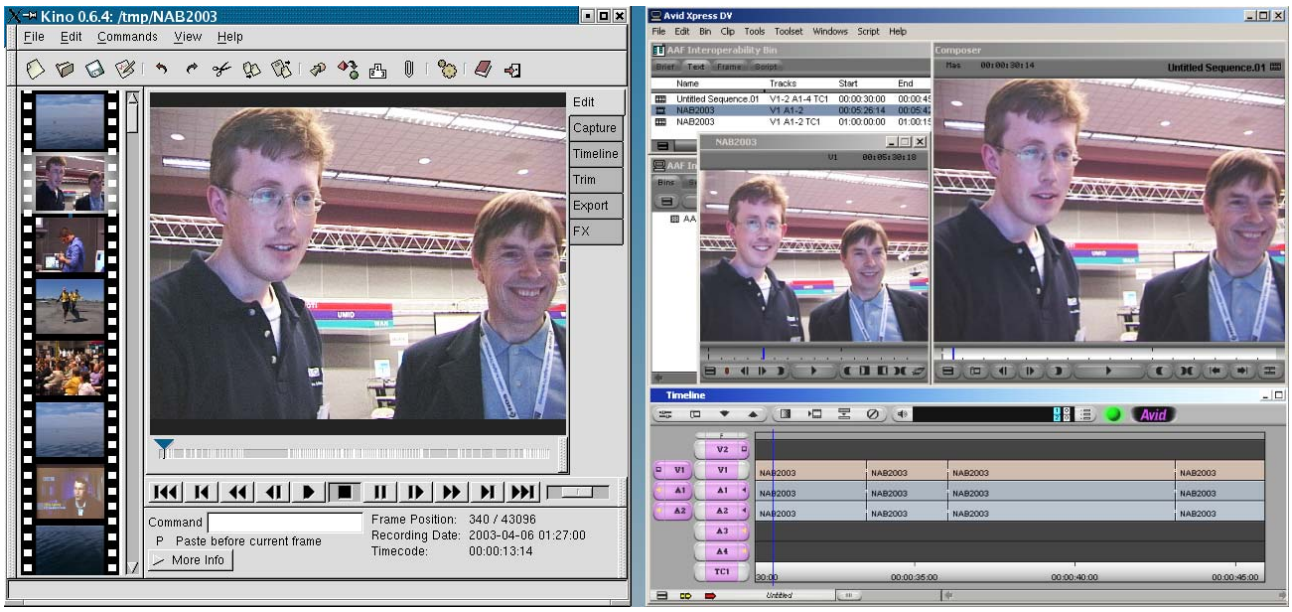

Figure 6 – Screenshot of Kino composition (left), and the same composition after being imported into Avid Xpress DV (right)

BBC planned software development

In addition to previous contributions, BBC R&D are developing a number of other software applications and libraries planned for open source release.

- **aaftest**: An application to verify correctness of AAF files is being developed. Tests are implemented using an extensible scripting language.
- **DV codec**: Building upon the work used to develop Kino's AAF-export functionality, a DV AAF plug-in codec is under development. It is hoped that the completion of a DV codec reference implementation will accelerate support for AAF in DV editing applications.
- **MPEG-2 codec**: The implementation of an MPEG-2 video codec providing an AAF plug-in codec API is under development. Real-time MP@ML video encoding and decoding capability is anticipated using libmpeg2 (11) and MJPEG Tools (12).
- **aafplayer**: A cross-platform multimedia application for playing AAF files with embedded essence. Development has concentrated on playing contiguous essence streams embedded in AAF files. Future work will conform compositions containing cuts-only transitions.

MXF open source software

MXFlib (supported by BBC)

MXFlib (13) (14) provides a C++ API for reading and writing MXF files. The software currently finds MXF partitions, reads metadata from any partition, traverses references and reads index table segments.

MXFtest (supported by BBC)

MXFtest (15) is an MXF file verifier. The software consists of a test engine which runs tests according to a test script. The implemented tests currently check MXF partition packs, the MXF Primer, referential integrity and the existence of all required and best-effort properties.

MPEG-2 open source software

The number of open source MPEG-2 implementations available is significant. The choice of those reviewed here was guided by the extent of usage in popular open source non-linear editors, DVD authoring packages and video processing tools. The key features of the resulting four open source packages chosen are compared in Table 1.

	Open source software package			
	MSSG	**libmpeg3** [16]	**libmpeg2**	**MJPEG Tools**
Library API	No	Yes	Yes	No
Supported profiles	MP@HL	MP@HL 422P@HL	MP@ML	MP@ML
Language & optimisations	C	C, MMX, 3DNow!	C, MMX, 3DNow!, SSE, ALTIVEC	C, MMX, 3DNow!, SSE
Codec operations	encode, decode	decode	decode	encode
Platforms	GNU/Linux, Unix, Win32	GNU/Linux, Unix	GNU/Linux, Unix, Win32	GNU/Linux, Unix

Table 1 – Key features of open source MPEG-2 implementations

CONCLUSIONS

AAF and MXF will be crucial formats for interchanging material and associated metadata between applications and platforms, as the move to distributed programme-making tools gathers pace. In addition, by drawing on a common, standardised, data model, the two formats work well together. The data model is scalable in the sense that it models the intrinsic manipulation and assembly of the programme-making process. This enables the model to describe complex real-world programme-making scenarios.

An advantage of describing the material sources is that the type of compression (if any) may be signalled. In a software platform, this allows applications to handle multiple types of essence compression reliably.

Specialised MXF OP-Atom provides a specification for editing applications from different manufacturers to access common material on shared storage, removing the inefficiency of an import process.

A considerable amount of open source software is now available in support of AAF, MXF and MPEG-2. The value of open source software to promote development of these formats is significant.

REFERENCES

1. Brightwell, P. J. and Tudor, P. N., 2000. A distributed programme-making environment using IT-based technology. Proceedings of 2000 International Broadcasting Convention. pp. 540 to 546.

2. SMPTE 335M / RP210, Metadata Dictionary Structure / Metadata Dictionary Registry of Metadata Element Descriptions

3. Pro-MPEG Forum. http://www.pro-mpeg.org

4. AAF Association. http://www.aafassociation.org

5. AAF Software Development Kit. http://sourceforge.net/projects/aaf

6. SMPTE 330M. Unique Material Identifier (UMID)

7. Quicktime. http://developer.apple.com/techpubs/quicktime/quicktime.html

8. Microsoft Windows Media. http://www.microsoft.com/windows/windowsmedia

9. Brightwell, P. J., Dancer, S. J. and Knee, M. J., 1997. Flexible switching and editing of MPEG-2 video bitstreams. Proceedings of 1997 International Broadcasting Convention. pp. 547 to 552.

10. MPEG Software Simulation Group. http://www.mpeg.org/MPEG/MSSG

11. libmpeg2. http://libmpeg2.sourceforge.net

12. MJPEG Tools. http://mjpeg.sourceforge.net

13. MXFlib. http://sourceforge.net/projects/mxflib

14. freeMXF. http://www.freemxf.org

15. MXFtest. http://sourceforge.net/projects/mxftest

16. libmpeg3. http://heroinewarrior.com/libmpeg3.php3

ACKNOWLEDGEMENTS

The contribution of Philip de Nier and Joseph Lord to the work described in this paper is gratefully acknowledged.

The authors would like to thank the BBC for permission to publish this paper.