

Operational Patterns... the MXF Flavors?

ERNESTO SANTOS

MOG Solutions, Portugal

This paper was first presented at the NAB2007 Broadcast Engineering Conference in Las Vegas, Nevada

ABSTRACT

MXF, the Material eXchange Format from SMPTE, is today a huge success. The market was eager for a standard file format that would serve as a basis for the new file based production paradigm, and MXF delivered precisely that.

As a result, the market is being flooded with MXF products, each using MXF in the configuration that suits best that product's application. For this reason, questions such as "which MXF flavor is this?" or "whose MXF is this?" are becoming quite common. And whenever these questions come up, so does the expression "MXF Operational Pattern".

This document discusses MXF Operational Patterns and other MXF technicalities, that relate to the way the tools in the MXF standard can be configured. These configurations usually end up interpreted by newcomers as MXF flavors. The document points out strengths and weaknesses of these flavors, and provides guidance on the building of MXF systems, as more complex flavors (with higher level MXF Operational Patterns) start being used.

The paper further explores the integration of different MXF flavors into the workflow, therefore defying the sometimes established notion that you must choose one and only one MXF Operational Pattern to work with.

SETTING THE SCENE

Every time a new technology arrives in the market, it takes its time to settle down. The deeper the impact on the everyday operations of the potential users, or their strategies for the future, or both, the more work needs to go into the way that technology is presented, and the way products and services are placed in the market.

This means that the detailed technicalities need to be refined and knowledge needs to travel from the people developing the systems, to the people marketing solutions and, of course, to the potential users of those solutions.

So what is the impact of MXF on everyday operations and strategies for the future? It could hardly be any deeper. The MXF file format enables the carriage of both the audiovisual material and related information, usually referred to as Metadata.

Metadata ranges from structural information of the material such as compression settings, to geo-localization information, to general descriptive information, including transcripts of the Content. This is why MXF represents more than a file format. It is a technology that enables the gathering of crucial information, as the audiovisual material travels through the workflow. It is therefore the foundation technology driving the adoption of Information Technology in the Professional Media market.

This also means that we are witnessing the convergence of these two different technology fields (IT and Professional Media market technologies, such as Broadcast technology), that evolved mostly separately over several decades. MXF is providing the vehicle for experts of both fields to meet half way, and its success proves the surprising speed at which this convergence is taking place.

Having the technical people meeting half way and understanding each other is a fantastic first step. However, their messages need to be refined towards a higher level of abstraction so they can be understood by potential users of the technology. These users don't know, don't need to know and don't want to know the technical details. They need just the right amount of information that will allow them to make strategic decisions for their businesses.

NEW KID ON THE BLOCK

Now, if you are a placing a new product in the market, you need to take into account:

- the current world view of your customers;
- where they want to go;
- the route they will take to get there.

The current world view of a lot of users is still based on conventional technology, such as tape/film based workflows. Most of them, if not all, have already demonstrated they want to move towards a new paradigm of file based production, management and distribution. This made it quite easy for manufacturers and service providers to figure out that MXF file based technology was the way to go. This is why you see MXF support popping up all over the place in the professional media market today.

Therefore, the key question is: which route will users take to get there?

So far in this document, the professional media market is being pointed out as the market for MXF... That is quite broad. That's because MXF was thought up to apply to virtually any scenario where professional audiovisual Content is being produced, managed and distributed.

This implies that there is a huge variety of potential MXF users, with different applications and different business models. And even those users that compete within the same application area, will most likely have different strategies and operational models. After all, they have to differentiate from one another in a world that becomes more demanding each day on the multimedia products quality, cost and time-to-market.

Strategies and operational models, in the end, map down to the workflows inside of an organization. Therefore, MXF had to be thought up as a configurable technology able to adapt to different workflows. And since strategies and operational models evolve over time as new ideas come along, MXF also had to be thought up as an extensible technology.

EXTENSIBILITY

A tutorial on MXF is outside the scope of this document. The reader should refer to [1] for detailed information on MXF, [2] for a discussion on how MXF integrates with related technologies such as AAF and XML, [3] and [4] for the entry points into the MXF Standard's document suite itself.

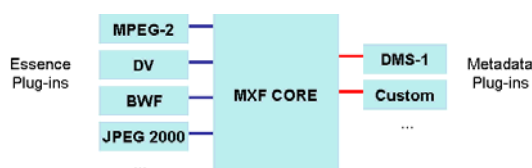


Figure 1

For the scope of this document, it suffices to say that the MXF standard defines a core set of tools which include the way the bits are encoded within the file, the way Metadata structures are logically organized, the way audiovisual material can be indexed for efficient random access... among other.

In addition, recognizing the need for extensibility, MXF is designed around a plug-in mechanism. The basic idea is that MXF is a wrapper of the audiovisual material. Therefore, if the material is compressed in MPEG-2 for example, that is how it will be represented inside the file, and that is the format that you get again once you unwrap the file. Then, for each format (MPEG, DV, JPEG2000, BWF ...) the MXF document suite includes one additional document that tells you how to wrap that format efficiently in MXF. This additional document will also tell you how to represent compression settings (for compressed formats) in the MXF Metadata, and how to index that specific format.

Therefore, for each new format that comes up in the future, you just need to write down a document using this extensibility mechanism, and plug it into the MXF document suite.

The same holds true for Metadata extensibility. At the MXF core, a structural Metadata model is defined that lets you know which clips¹ you have in the file, how many tracks, their bitrates, timecode information... Then, a plug-in mechanism allows you to extend this Metadata with additional descriptions, following Metadata models that you can design yourself. It also enables you to frame accurately synchronize your Metadata with the audiovisual material you are describing. These Metadata extensions are usually called Descriptive Metadata.

SMPTE DMS-1 [4] is one example of a Metadata model plugged into MXF, using the above mechanism. In this case, it is a standard Metadata model, however you can use the same mechanisms to plug in proprietary Metadata models as well.

ADAPTABILITY

The described plug-in mechanisms provide proven support of MXF extensibility. However, all these mechanisms imply a complex MXF core. This means that decoders would have to be very

¹ For the purpose of this text, clips should be considered a sequence of contiguously stored audiovisual material.

complex, to support all possible variations of MXF configurations thrown at them.

So how do you adapt MXF to application needs? The main aspects to be taking into account in MXF are the audiovisual material encoding, the Metadata model being followed, and the configuration of the whole.

Audiovisual material encoding is not something defined by MXF. And its adaptation to application needs is also not a new subject. There are several possible encodings for uncompressed material and several compression systems as well. The concept itself of compression was once new in the market as well but, although there is always evolution, the concept settled down already. There is currently extensive know-how on the criteria you need to follow to decide on which compression system, or which uncompressed representation to use. There are also several well established parameters you may tweak to get to the expected results.

Therefore, even when a new compression system arrives in this field, you can always plug it into this knowledge framework and compare with existing systems. An example is perceptual quality measurements of new codecs against the ones in the installed base.

Then we move to Metadata models... and that is a new world on itself. More than a few descriptions on a file, this is the representation of knowledge and processes, or in the end, the capturing of information flow at its highest level. A detailed discussion on Metadata models and their adaptation, although very much related to the subject of this document, deserves a document on its own and will therefore be considered out of the scope of this discussion.

So finally we need to take into account that, having all components in a system understand all possible variations of MXF configurations, would make such a system extremely complex and therefore extremely expensive.

Therefore, MXF was thought up with configuration control knobs. These are parameters that enable you to “speak MXF” with more or less complexity, and enable you to signal which level of complexity you are using.

You will often come across several of those parameters in the MXF lingo: Clip Wrapping, Frame Wrapping, sparse or full Index Tables, interleaving and multiplexing... these are just a few of the control knobs that you can tweak in the MXF toolset to configure it.

The most popular control knob these days is the Operational Pattern. And there is a reason for this. The Operational Pattern control knob is the highest level one. It tells you if the file has one clip or more. And if it has more, it tells you upfront what your decoder should expect in terms of the complexity of the relationship between those clips.

OPERATIONAL PATTERN

For the purpose of Operational Pattern classification, you need to regard an MXF file as a file that holds clips of audiovisual material that you want to play out in some specific way.

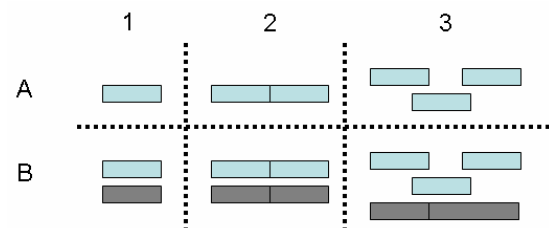


Figure 2

You may want to play-out (notice the relation to the numbering in Figure 2):

1. the full duration of one clip;
2. a sequence of full clips;
3. a sequence of selected portions of clips.

In addition, at each point in time you may want to be playing from:

- a. one clip;
- b. more than one clip.

In any of the cases above, a clip might mean just picture information or just sound information. However, a clip may also mean picture and sound information if both are stored together inside the file².

An additional level of complexity may be signaled in MXF Operational Patterns. You could see this as row ‘C’ that could be added in Figure 2. The row was not added because the semantics of this ‘C’ row follows a concept that is quite different from the one in ‘A’ and ‘B’.

For rows ‘A’ and ‘B’ the decoder is being told the (single) way to play the clips. In Operational Patterns of the ‘C’ kind, the decoder can choose from 2 or more ways to play-out the file.

² If a clip has both picture and sound information, this will mean that the picture and the sound samples are interleaved within the file.

So for OP1C, for example, you can choose from different versions on how to play the file, each individually following the rules in OP1A or OP1B. And a similar rule applies to OP2C and OP3C. A simple OP1C example would be a file holding both a full quality version of the material and a proxy³ version.

All the above Operational Patterns are called Generalized Operational Patterns, since they represent a general model for application scenarios.

The industry identified the need for additional Operational Patterns, targeting specific scenarios. In MXF terminology these are Specialized Operational Patterns, and OPAtom is a well know example.

OPAtom was designed to be the simplest way of placing single track material within MXF. Therefore, an OPAtom MXF file can have a picture track or a sound track; but not both. You can then use MXF Metadata to link together files that are part of the same clip. To have both in the same file, you should use OP1A.

Summarizing, classes of applications were thought up and patterns were identified on the way that these applications would need to organize their MXF files. This way, through the Operational Pattern, an MXF file can signal the decoder its “class” and therefore the level of complexity within. This way decoders can be built to support different levels of complexity, depending on the application field.

MXF CONTROL KNOBS... A THREAT?

All the other MXF control knobs quickly mentioned above, and quite a few others not mentioned here, are also important. However, the knowledge on their meaning and implications didn't travel from the people coding the bits to the general public knowledge, yet. So you can regard the Operational Pattern control knob as the first one to surface...

Is there a threat in there? How long will these other control knobs take to surface? And when they do, how complex will this get?

Actually if you are already familiar with MXF, you know that the fact that MXF is being widely

deployed, is leading its usage patterns to be identified by a number of users and manufacturers working together at SMPTE and other forums. This is leading MXF experts towards an understanding of the best way to surface those other control knobs. Nothing new, just business as usual, that is: learn from your customer feedback, and improve technology.

If you are reading this document as a newcomer to MXF, then you should see this as an opportunity. You need to start using this technology today. And you have the opportunity to feedback to manufacturers on your expectations for the functionalities of the next products' versions, so the control knobs can surface taking your input into account as well.

FLAVORS

The discussion in this document finally got us to the current state in the market. The concepts above should provide some insight on the why several manufacturers in the market, even within the same application areas, are deploying slightly different MXF configurations.

Some of these turned the Operational Pattern control knob to OPAtom. Others turned the control knob to OP1A. Others still, identified they needed to give their users the freedom to handle the control knob, so they let the user choose the Operational Pattern.

Now, here comes a problem: you look at a product's datasheet and see MXF; then you look at another product's datasheet, and read MXF as well. So, you can just connect them and it will all work, right? Not necessarily. As discussed above, internal control knobs may be set differently. And as discussed above, there are very good reasons for that from a manufacturer point of view, which make all sense.

Nevertheless, there are also very good reasons for users to become upset. After all, if there are differences, that is, if there are flavors, people should find a simple way to explain those differences. This is where we come back to the previous discussion, on the need to refine the technicalities of new technology into a higher abstraction that is understandable by the general public.

Manufacturers understood this and steps have been taken to make the users aware of the fact that MXF is not monolithic. Users are now being made aware that we are talking about a toolset, and products using this toolset expose control knobs. And that is

³ Proxy is one of terms commonly used to mean a lower quality version of the content that can be handled with fewer resources. E.g: highly compressed and reduced resolution version for playback over the Internet.

why today, even newcomers to the MXF world, immediately learn a few technical terms like OPAtom and OP1A. Datasheets are now starting to include these terms as well...

A side note: which other control knobs will come up next? Probably Index Table related control knobs will surface in the near future... but the real show will be the coming of the Metadata control knobs.

Metadata handling was the goal of MXF from the beginning. That is where the real power of MXF will be unleashed. Several pioneer projects are being deployed today worldwide setting the scene for this. However, as stated above, that is a world on its own that deserves a number of documents dedicated to it so we will keep it outside the scope of this document.

OP1A AND OPATOM... IS THAT IT?

Most products in the market today use either OP1A or OPAtom, and it is not surprising that these "OPs" are so widely used. These are the simplest MXF configurations, and it was perfectly expectable that these would be the first ones to reach the market.

A number of projects are popping around the globe starting to use the higher level Operational Patterns described above. Of course that represents a dramatic increase in functionality, but also an increase in complexity. Therefore, manufacturers need to be very careful in designing products from solid user feedback.

This is also why most products out there supporting all Operational Patterns are development tools, such as the MOG Solutions MXF::SDK⁴. And projects are using those tools to implement pioneer deployments that gather feedback on users' needs, and feed those into the design of the next generation MXF products.

WHICH "OP" TO CHOOSE

So given all this, which Operational Pattern should you choose for your system?

The MXF purely technical answer to that is that you shouldn't choose an Operational Pattern. The Operational Pattern should be a consequence of your design choice for a system.

Each organization has its own specific internal processes. Some always treat picture information and audio information separately, and have deployed systems to keep track of the location of the different assets, and the relationship between them. In this kind of system, maybe you want to make sure you keep roughly the same processes when switching to MXF, so that your staff can handle the change easier. So if you conclude from your analysis that you want to keep the picture and sound information separately, you will probably conclude that OPAtom is the way to go. And you will then gradually improve the linkage and synchronization between the different assets, by using mechanisms available in MXF Structural Metadata that enable you to link the OPAtom files together.

A notable example here is the Digital Cinema distribution. There is great tradition in the Motion Picture industry of separating the treatment of picture and sound information. Therefore, it is not surprising that the DCI specification [5] adopted OPAtom with separate MXF files for picture and sound information.

On the other hand, suppose the staff in your facility is used to handle the Content contained within a package (such as a tape containing all the video and audio to be used). In this case, the least disruptive way to introduce MXF to your facility, would probably be to keep the files with both picture and sound information... a kind of virtual tape.

Therefore, in this case, you would probably want to have MXF files which interleave picture and sound information, which would probably lead you to the OP1A side⁵.

A simple example here would be a department that keeps a short term archive of the rushes coming in from the camera crews, and that doesn't want to invest into file management databases to keep links between assets. They just want to have a file repository where each file contains the full package. OP1A would work as the full package in this case.

Now suppose you need to archive your material using MXF, and you would like each file to have both a full resolution version and a proxy version. Well, in this case you would conclude you need to

⁴ Developed in collaboration with IRT – www.irt.de.

⁵ MXF experts will probably note at this point that you could also perfectly have separate sound and picture with OP1A files... please note though that this document is all about guidance, general trends and keeping it on the mainstream. You always have the choice to go for specialized solutions, of course.

have multiple versions of Content in the same file, and it would probably lead you to OP1C.

One final example: suppose you want to send to a dubbing facility an MXF file, with some English-spoken Content for which you need an additional sound track with the dubbing to the Portuguese language. If you want to have the new track interleaved with the picture information, and therefore immediately optimized for streaming, this would mean a re-wrapping scenario for the dubbing facility, and you would probably get back OP1A.

On the other hand, if you are going to keep the remote facility technical requirements down, and just use MXF as the container that holds all the assets so these don't get lost, then you probably want to get back an OP1B file.

The way it could work is:

- Dubbing facility receives the OP1A;
- Dubbing facility appends the soundtrack to the MXF file (appropriately encoded using the MXF encoding mechanisms, of course);
- Dubbing facility performs some MXF OP1B "magic" in the Metadata of the MXF file, so that when you play it out, you listen to the new Portuguese soundtrack rather than the original English one... However the original English version will still be intact in the file, in case you need to reuse it.

ADAPT MXF TO YOUR NEEDS

The previous section focused on the purely technical answer to the question on which "OP" to use. Of course in reality we need to take into account more than that.

In the end, when you design a system you will be using products in the market. So, you may want OP1C, or OP2B or some other...but if the products you need support only OP1A and/or OPAtom, than you have to take that in consideration.

Does that mean you need to live with what you get from those products? Absolutely not. When a new technology enters a market, initially users have to live with what they get in the product box. As the manufacturers get more feedback and learn more about users' expectations, the ones that win the race are the ones that adapt the technology to the users needs... not the other way around.

So the key here is: you should adapt MXF to your workflow, not adapt your requirements to what specific MXF products give you. How do you do that? MXF adaptation processes. Different MXF

configurations will work better on different sections of your workflow. So you need to take into account that you will need processes in your workflows that automatically turn control knobs for you to adapt MXF from one operation to the next.

Just look at how it was done with the compression systems. Of course initial products were deployed supporting very specific compression operating points. Nevertheless, soon a full industry was born providing converters that let you choose which format to use at each stage.

To capture video you probably use something like MPEG-2 I-Frame 50Mbps or DV 25Mbps... You distribute it to people's homes using MPEG Long GOP... You store proxies for browsing using MPEG-1 in your media asset management ... and probably even post a few clips on your company's website using MPEG-4, for example...

Conversions are all over the place in your workflow already, adapting your Content to each specific section of it. Doesn't it make sense that similar adaptations apply also to the MXF wrapper⁶?

SO WHY DO WE NEED MXF?

I guess the obvious question after that would be: why do we need a standard called MXF if you need to do adaptations anyway? Why not just different formats at each stage then? Because no matter which conversions, for example from an OP to a different OP, at both sides of the converter you still have the same technology: MXF. And there are a number of MXF tools⁷ that you will always be able to use, irrespective of the Operational Pattern to perform defined sets of operations with the files.

Additionally, on the manufacturer side, it means engineering teams can concentrate on developing and researching using one base set of tools (the MXF standard document suite) rather than having to reinvent the wheel for each specific step in the workflow. This means lower costs for the manufacturer, improved product quality and

⁶ If you think about it, transport/wrapper format conversions already occur all the time (think of current workflows mixing tapes, SDI connections, satellite links...).

⁷ An example is a MXF player and Metadata editor like MOG Solutions theScribe. Some of its functionalities apply only to specific "OPs". However, something like playing and Metadata editing applies to any Operational Patterns you throw at it.

reduced time-to-market, with the obvious advantages this implies for users as well.

Finally, keep in mind that a major goal of MXF is to provide the framework for Metadata handling. The model of constant adaptation throughout the workflow, always using MXF, enables you to deploy the foundation framework you will need to tunnel your Metadata through. You can hardly do this if you have different formats across the workflow.

This model also prepares you for the next step, where your system becomes more comprehensive. You will then get to a stage where you identify the need to plug-in Metadata adaptation as well, at specific stages of the workflow.

SAMPLE SCENARIO

Before we conclude, maybe it will be helpful to the reader to consider a sample scenario that uses some of the examples in the text above, and where it becomes clear how MXF interoperability doesn't mean that every box in a system needs to understand every MXF flavor. It just means that the same MXF toolset is adapted to fit the needs of different points in the workflow.

In this simple case, and since we have been focusing on the Operational Pattern control knobs, adaptation tools being deployed are Operational Pattern converters integrated, in some cases, with compression codecs. Don't forget though that you can have adaptation tools automatically turn for you a lot more MXF control knobs than just the OP control knob.

In the figure, Content can get in from different providers. Some deliver OP1A some deliver OPAAtom.

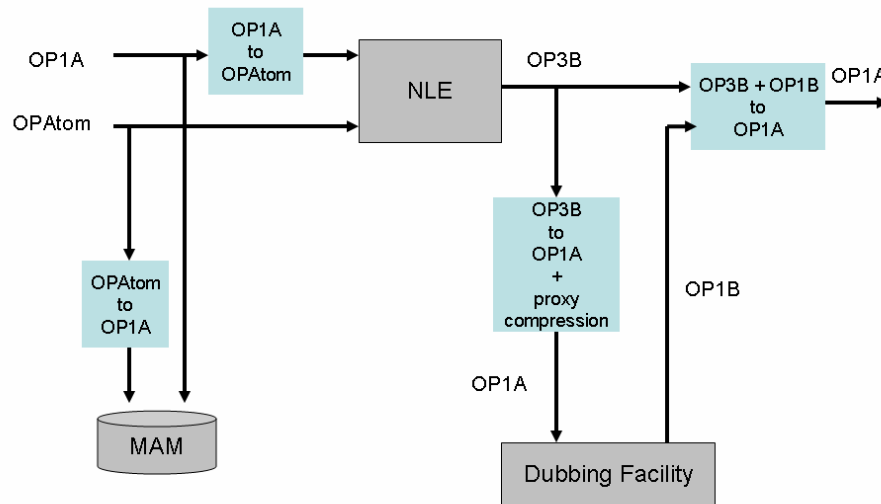


Figure 3

All input is stored in short term archive and, to simplify file management, all Content is kept as OP1A files.

The editors in this particular facility always treat video and audio separately, so all OP1A Content is split into OPAAtom files before entering that department. The output is a OP3B file, which is basically an MXF file which includes the editing list and all referenced clips.

The OP3B Content is delivered to a proxy production system, which transcodes all Content to a proxy version packaged as an OP1A file.

The proxy is sent to a dubbing facility over the Internet, where an alternative language audio track is created, and then added to the file making it now an OP1B proxy file.

Back at the post-production facility, the OP3B file is combined with the dubbed soundtrack of the proxy OP1B file to produce a final OP1A file, which holds the final product.

Notice that throughout this process you have Essence transcoding, just as you already have today. Notice that you can also have Metadata transcoding since different Metadata fields will be relevant at each point. As an example, maybe you want to have the source camera serial number and cameraman contact information on all rushes... however it probably doesn't make sense to keep that information on the proxy being sent to the dubbing house.

On the other hand, Metadata such as episodic information, rights information, transcripts of the soundtracks, among other, may be very useful to keep across the workflow in the different versions of the Content. And you can tunnel this Metadata throughout the all system, because it is using the same MXF wrapping technology everywhere.

In the figure, you could easily make that Metadata travel through all the MXF paths and make it editable at any point you wish... While still taking advantage of the processes automatically tweaking MXF control knobs for you, to adapt that wrapping technology to most suitable configuration at each point.

CONCLUSIONS

MXF was developed to be extensible and to be adaptable. This enables this technology to be used in a variety of different scenarios.

Adaptability is achieved through control knobs that enable configuration of MXF for each scenario. Some of these control knobs are already common knowledge, with Operational Patterns getting the popularity prize there. Other control knobs will emerge, as the market gains more experience on this technology, and more projects experiment with higher level OPs and rich Metadata workflows.

This document seeks to project a image of a technology that is replacing a number of other technologies used across complex workflows. This document puts forward that it is not feasible to reach such an ambitious goal with a monolithic approach. Therefore, MXF should be regarded as a toolset that is expected to be constantly adapted throughout the workflow.

This means that not all blocks in the system need to support all possible MXF configurations, therefore simplifying their design and reducing their costs.

However, since a common toolset is being used, this enables new products to be developed and easily integrated in the systems. It also enables rich Metadata applications, the principal goal of MXF, to tunnel their Metadata throughout the system.

Manufacturers need to do a better job at outlining their products functionality and interfaces. They also need to provide more Metadata functionality on their systems, so that the tunneling of that Metadata through complex workflows, with constant adaptations as in the scenario above, clearly demonstrates the functionality MXF aims to provide.

Users need to be more proactive. Yes... it is true that big-budget projects can change complete workflows and deploy spectacular systems. But small-budget projects can't just wait to see what happens next. It is already a fact that MXF is the way to go. The only question is which route each user will take.

Results from other projects will for sure provide guidance on the route to be taken. Nevertheless, in the end, the technology needs to be adapted to each users needs, not the other way around.

Therefore, step by step, each user needs to proceed in the adoption of this technology, bringing in the adaptation tools being made available, and deploying these with a strategy of adaptation to their own way of doing business.

REFERENCES

- [1] Devlin, B., Wilkinson, J., Beard, M., Tudor, P. and Wells, N., 2006. The MXF Book. Focal Press.
- [2] Santos, E., 2006, AAF, MXF and XML... Putting it all together, IBC 2006 Conference Publication
- [3] SMPTE, 2004. S377 - Material Exchange Format (MXF) - File Format Specification. <http://www.smpte.org>.
- [4] SMPTE, 2004. S380 - Material Exchange Format (MXF) Descriptive Metadata Scheme-1. <http://www.smpte.org>.
- [5] Digital Cinema Initiatives, LLC, 2005. Digital Cinema System Specification V1.0. <http://www.dcinovies.com>