Multimed Tools Appl
DOI 10.1007/s11042-010-0513-3

Interoperable digital rights management based on the MPEG Extensible Middleware

Víctor Rodríguez Doncel · Jaime Delgado · Filippo Chiariglione · Marius Preda · Christian Timmerer

Abstract This paper describes an interoperable Digital Rights Management architecture promoted by the MPEG standardization group in its new standard known as MPEG-M or MPEG Extensible Middleware (MXM). The goal of this standard is to promote the packaging and reusability of MPEG technologies, and for this it specifies a software middleware platform and a complete set of APIs and protocols. These APIs allow uniformly handling digital content and developing generic multimedia applications, through a set of modules communicated with standard protocols. The MXM standard provides the necessary mechanisms to digitally manage the rights over Intellectual Property, the tools to protect the media and the means to grant the rights enforcement, besides a rich set of libraries to reproduce the content.

Keywords Middleware · DRM · Content management · MPEG-21

1 Introduction

On despite of the technology fragmentation, traffic of audiovisual content has been growing in the recent years at a striking pace. Both free and paid content circulation figures have been growing, but these numbers could have been larger had

V. Rodríguez Doncel (✉) · J. Delgado
Universitat Politècnica de Catalunya, Barcelona, Spain
e-mail: victorr@ac.upc.edu

F. Chiariglione
CEDEO.net, Torino, Italy

M. Preda
TELECOM and Management SudParis, 91011 Évry Cedex, France

C. Timmerer
University of Klagenfurt, Klagenfurt, Austria

Published online: 08 April 2010

a standardized technology been established. This was the opinion of a group of digital media experts, assembled by Leonardo Chiariglione in late 2003, which published the Digital Media Manifesto [15]. Its premise was that digital media development had not reached the full potential promised by the technology and that digital content creation, distribution and consumption could be boosted if digital systems were more compatible, policies less arbitrary and technology more neutral. The conviction was that a fair model could be traded-off and the interests of content creators, producers, distributors and consumers could be easily satisfied in an agreed model. The Digital Media Manifesto identified the need for a set of coordinated policies and technical actions. This led to the development of specifications for interoperable Digital Rights Management (DRM) platforms technically open to value-chain players and for heterogeneous end-user devices.

The Digital Media Manifesto was the starting point of the Digital Media Project group, which immediately set out to make this idea true and started working on it. The Digital Media Project has been working since as an unaffiliated, geographically diverse, nonprofit group, and after 6 years it still remains active and continues to reach remarkable landmarks. Their most prominent outcome is the specification of the DRM interoperable platform [1] and the establishment of a worldwide Content Registration authority. An implementation of the interoperable DRM specification (whose first version dates back from 2006) has been given by the open source Chillout project.[1] Chillout is based on several MPEG standards—mostly MPEG-21 and MPEG-A—, and has been demonstrated during the Beijing Olympics (WIM TV trial)[2] in 2008.

As the DRM interoperable specification had relied mostly on MPEG technologies, the next logical step was to return the results to MPEG. Thus, the project to develop an interoperable platform was launched within MPEG aimed at establishing a new international standard. MPEG-M [11], formally known as ISO/IEC 23006 (Multimedia Extensible Middleware) is in an advanced state in the standardization process, and it comprises a suite of technologies with the purpose of enabling the easy design and implementation of media handling value chains, fulfilling the Digital Media Manifesto vision. In particular, MXM (MPEG Extensible Middleware) defines an architecture comprising MXM APIs on top of which MXM applications run on MXM devices. This paper serves as an overview of the most relevant features of MXM emphasizing its conception of DRM.

2 Related work

The MXM assumption is that media-handling information systems can develop faster thanks to their homogenous basis and because of better interoperability with other systems, an idea quite different from proprietary DRM systems like Applet's iTunes, Microsoft Windows Media DRM or RealNetworks' Helix DRM. These systems are

[1] The Chillout Project (2009) http://chillout.dmpf.org.

[2] See http://www.wimtv.it/.

said to be closed platforms because content purchased for one of them cannot be reproduced in another (excepting ephimeral attempts like RealNetworks Harmony which allowed content from RealNetworks store to be played in Applet's iPods.). However, besides MXM, some other DRM platforms have been developed with interoperability at the center of the design criteria.

This is the case of the Coral Consortium, a standards initiative dedicated to interoperability of DRM systems gathering technology companies and content producers. While it also accepts the idea that DRM interoperability can be solved through standards alone, Coral reduces the development load to a minimum by adopting the notions of interoperability between DRM systems and conversion between DRM systems. Thus, Coral specified in 2004 a testbed system called NEMO (Networked Environment for Media Orchestration) in order to achieve interoperability among different devices, formats, networks, and types of services [9]. NEMO is a secure messaging scheme based on X.509 identity certificate and SAML security assertion standards, whose interoperability comes from wrapping existing systems into standardized services. Some NEMO based systems have been deployed with a certain degree of success, like Marlin or DReaM, the Sun's open-source standard for interoperable DRM.

Other open interoperable architecture is OpenSDRM, presented in [14]. It includes a set of components also similar to those of MXM, with the sole addition of the payment infrastructure, about which MXM is currently agnostic, and a major difference in the authorisation mechanism, which is performed only locally in OpenSDRM and may be remote on MXM.

The MIPAMS (Information Protection and Management System) system described in [18, 19] also represents an MPEG-21 based architecture, and its overall structure of trusted modules resembles that of MXM. The major difference with MXM resides in the effort of standardizing the *protocols* followed by the different devices. Standardising the payload of the information exchanged between applications increases interoperability among applications and potentially grants an easier replacement for newer implementations, at the cost of some extra overhead. Some of these protocols had been already standardized as part of ISO/IEC 29116-1 [8].

The last of MXM's precedents has been Chillout, one of the project major inspirers. Chillout is the open source implementation of the Digital Media Project (DMP) Interoperable DRM Platform which was launched before the MXM initiative started. Chillout's code was released under the Mozilla v1.1 license, it includes most of the components of the DRM system and has inspired many of MXM, but APIs organization in MXM is superior including a metamodule for the fast development of applications.

3 Overview of MXM

MXM supports different business models and aims at satisfying the many parties which in one way or another may interact with media. This section reviews the media value chain (Section 3.1), the main architecture elements (Section 3.2) and the utility it can provide to different kinds of users: content creators, end users, distributors, broadcasters, graphic application users (Sections 3.3 to 3.7).

3.1 The media value chain elements

One of MXM's objectives is to provide a middleware on top of which applications supporting most of the links in the value chain can be built and deployed. The media value chain is the life cycle of content since it is conceived until it is consumed. Different business models involve different actors and different steps in the evolution of the message, and thus while distributing music on a CD needs a 'producer' and a 'retailer', a TV news program needs a 'broadcaster'. In its most simplistic view, it is a matter of information communication between a source (the content creator) and a receiver (the end user), with a channel, a content, a codification etc. In practice, there are a number of intermediaries and message transformers in between, and MXM tries to provide them some help. These abstract elements in the transaction include:

- *Content sources*. MXM provides access to content creators for encoding algorithms to represent audio, still image, video and graphics. Once encoded, enriched and structured in Digital Items (what is described below), MXM supports content registration by specifying protocols to communicate with content registration agencies, as well as those to store content on remote repositories relying on existing delivery methods. Finally, it provides content authors with the possibility to define the terms under which their work can be accessed, traded or modified. The group of content creators may include creators of original works (authors), adapters who derive new works and interpreters who perform different versions of a work, all of them generating new value in the value chain and providing different forms of content.
- *Content handlers*. The group of intermediaries between content sources and content sinks includes content distributors, retailers, publishers, device manufacturers, Collective Management Societies, libraries, rating agencies, etc. MXM provides them with the means to exchange, share, trade, etc. digital goods among themselves or with the end user in a controlled manner. Transferred content can be licensed and licenses can be issued, stored and authorised in a standardized way. Also, content can be adapted to the end-user needs, being this content adaptation also expressed in a standardized way.
- *Content receivers*. End users can enjoy different applications and services built on top of the MXM middleware, which facilitates media handling and rendering.
- *The message*. The Digital Item is the unit of information postulated in MPEG-21, the multimedia framework, and so it is in MXM. MXM provides the needed software to handle Digital Items among the users in the whole value chain.
- *The channel*. MXM provides help at transmitting media (e.g. as streaming, downloading or through P2P channels), but also the means to do it in a secure way. Every communication channel between two devices is established after they have mutually authenticated, and registries of devices and domains may facilitate this even more.

3.2 MXM architecture elements

The MXM standard is organized in four public documents, describing the architecture, the APIs, the reference software and the protocols:

- MXM architecture and technologies [5]

– MXM application programming interface [6]
– MXM conformance and reference software [7]
– MXM protocols [8]

The fundamental part for someone approaching the standard for the first time is the first one, where the whole design is presented. In this document, the MXM Device is described as a platform able to run one or more MXM Applications. These applications run on top of the Operating System, and additionally may exploit a number of MXM Engines enabling access to the MXM technologies and protocols.

Each MXM Engine is backed by an MXM API (defined either in Java, C++ or both), and the interoperation between MXM Engines is well defined through the protocols. Protocols exchange a well defined set of XML messages, in practice delivered through Web Services.

The MXM standard is mainly concerned with the interfaces of the MXM Engines, named MXM Engine APIs and the interface to a master Engine called Orchestrator Engine. It also defines the API calls, but not how they may or should be implemented. The implementation will actually be different for different computing platforms (either hardware or software, like the OS) (Fig. 1). It is likely, however, that a sample model implementation is given as a Reference Software by the standard itself. In the current specification, MXM defines 18 engines, which are listed here:

– *Orchestrator Engine*, a special engine capable of invoking predefined sequences of calls in the others engines and capable of providing a simpler, unified interface to a priori known application domains.
– *Media Framework Engine*, supporting video, audio, image and 3D graphics handling, it provides methods to create (encode) and access (decode) the elementary streams.
– *Rendering Engine*, providing the access to hardware graphics acceleration, input device handling and its abstraction.
– *Metadata Engine*, for creating and accessing audio, still image, 3D graphics and video metadata.
– Content Search Engine, implementing the MPEG Query Format [4] methods to access content repositories (the content provider device).
– *Digital Item (DI) Engine*, concerned with the Digital Item Declaration, the Digital Item Identification and all related operations to create access and edit Digital Items.
– *Content Protocol Engine*, with the procedures to identify a content item from a content identification device, to access it and store it from/to a content provider device and to authenticate it completely or in parts.

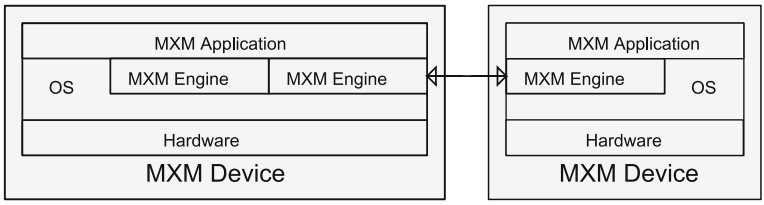

Fig. 1 MXM components model

- *MPEG-21 File Engine*, concerned with the serialization and access of Digital Item as files (with the ISO/IEC 21000-9 MPEG-21 File Format).
- *Digital Item Streaming (DIS) Engine*, concerned with the serialization and access of Digital Item as streams, operating over ISO/IEC 21000-18 Digital Item Streaming data structures [16].
- *Digital Item Adaptation (DIA) Engine*, providing the methods to parse, access and create information contained in usage environment description elements (context information needed to perform content adaptation).
- *Event Reporting (ER) Engine*, to track every relevant action in the system, as well as to create Event Report Request and Event Reports conforming to ISO/IEC 21000-15 [17].
- *Security Engine*, providing cryptographic algorithms, digital signature and the tools to achieve trust in devices. Note that devices interoperate with each other after having mutually authenticated themselves, by exchanging certificates provided by the Certification Authority.
- *Media Value Chain Ontology (MVCO) Engine*, with the methods to access the value chain ontology and manage Digital Items and users in conformance with the IP value chain model.
- *Rights Expression Language (REL) Engine*, capable of parsing and editing MPEG-21 REL licenses, according to the standard and its extension with profiles.
- *License Protocol Engine*, declaring the methods to access store and revoke licenses remotely in a license provider device.
- *Domain Engine*, implementing the methods to exchange information with the domain management device.
- *Intellectual Property Management and Protection (IPMP) Engine*, classes to create and access IPMP data structures, which describes the mechanisms and tools for the protection of Digital Items.
- *IPMP Tool Protocol Engine*, providing the methods to access an IPMP Tool Body, i.e., how to access an implementation of protection algorithms.

3.3 MXM for a content creator

Content creation is present in every DRM platform. The first step in the chain—and without which there is no chain—is the creation of digital objects. These digital objects usually have a precise format, and the step of creation may actually be content injection from other existing databases. Accompanying the digital objects, usually metadata is present so that searches can be possible as well as an easier management. Digital objects may be protected so that they travel safe and even though they are reached by a malicious party, data is still safe. Examples of content creation tools in the different platforms are the Helix DNA Producer application of Helix DRM, the Microsoft's Windows Media Encoder (which comes along an API to create protected content) or the Marlin's Bento4 Packager to create content.

In MXM, content creation is essentially the creation of MPEG-21 Digital Items, which encompass the insertion of both resources and metadata in a uniform way for any media type. The creation of content in MXM may involve the following steps: (i) creation of the media resources (audio, video, graphics, etc.), (ii) identification of the media resources, (iii) description of the media resources by adding metadata,

(iv) definition of the protection information for the various parts of the content item, (v) protection of the selected elements by means of algorithms named DRM Tools, (vi) identification of the protected elements, (vii) definition of the rights information for the content item or parts thereof, (viii) storage of license templates onto special servers capable of issuing licenses upon request, (ix) creation of the Digital Item containing all the information, and (x) identification of the new content item and its exposure to the value chain model managed in the Role Verification Device (which implements access to the MVCO Engine methods).

This sequence of steps, partially implied in Fig. 2, clearly shows possible relationships between applications involved in the process of creating content, which are typically run by different companies and individuals and require different skills.

3.4 MXM for an end user

Content is consumed in a variety of ways through applications which render or reproduce content, the end user devices. Typical end user devices have secure storage, and execution and consumption environments where unauthorised parties cannot access keys and status information. Besides proprietary end users devices (iPods, Zunes or Helix DNA Players), applications can be built on top of APIs, but they are not free (like Microsoft) even when they are said to be open (Sushi Marlin Client SDK for creating client-side Marlin DRM applications).

MXM offers instead all the elements for developing DRM client applications. Figure 3 shows some possible MXM devices involved in the content consumption. The end user device can be any multimedia player, streaming receiver device or set-top box. While the perception of the end user is limited to the single device he uses, in practice the operations released when a rendering operation is performed are complex and involve some of the surrounding blocks of Fig. 3. For example, if an end user device belongs to a domain, and this domain has been licensed to obtain certain content, the license retrieval will imply establishing a protocol with the domain management device, in a process that is transparent to the user.

The complexity of the operations in the previous example can even be transparent to the end user device developer, by virtue of the Orchestrator API (not shown in the figure). Thus, the implementation may invoke either directly the MXM APIs methods (e.g. the Domain Engine, the License Protocol Engine etc.) or just simply the Orchestrator API, which will take charge of all these calls. Once the rules of

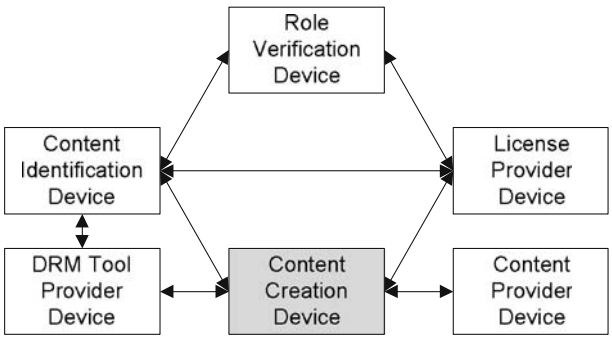

Fig. 2 Devices involved in the content creation

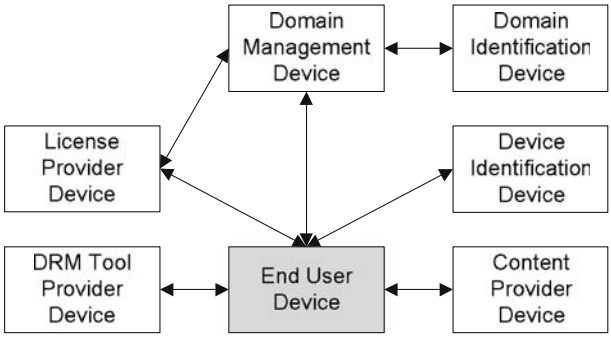

Fig. 3 Devices involved in the content consumption

consumption are established, the Content Provider Device gives access to the media through an API transporting flat (uncompressed) representation of content.

The end user device communicates with the content provider device, the license provider device and the DRM Tool provider device in order to retrieve content, license and decrypt algorithms, what will be needed to render, store or transfer the Content.

3.5 MXM for a mobile end user

A mobile user is consuming or even producing digital media in various contexts. Other DRM platforms also offer their SDKs for developing mobile applications, but none is fully MPEG based. As MXM technology abstracts the hardware layer, it is conceivable having MXM applications running on mobile devices. However, and besides the procedures described in the previous paragraphs, specific provisions have to be taken due to the heterogeneous nature of the physical devices, and more in the general, the context.

In multimedia systems, the context is usually referred to as any information that can be used to characterize the situation of an entity (e.g., user, terminal, network, etc.). MPEG-21 Digital Item Adaptation provides means to describe this context through its Usage Environment Description (UED) tools [20]. The UED is clustered into properties which allows for a comprehensive description of the user, terminal capabilities, network characteristics, and the natural environment where the digital media is ultimately processed (i.e., consumed or produced).

The MXM DIA Engine provides means to create and access (i.e., parse) information pertaining to the usage environment context and, thus, enabling device-independent access to digital media regardless of the geo-location, service provider, network provider, device manufacturer, etc. In particular, the MXM DIA Orchestrator Engine provides means to adapt the digital media—possibly described by metadata and accessible through the MXM Metadata Engine—according to the UED provided through the actual DIA Engine and by utilizing a Media Framework Engine which performs the actual adaptation (see Fig. 4).

Additionally, and thanks to the MXM protocols, interoperability at the signaling level is guaranteed. That is, the Request Content part of the MXM protocols provide means to communicate context information—among others—towards the

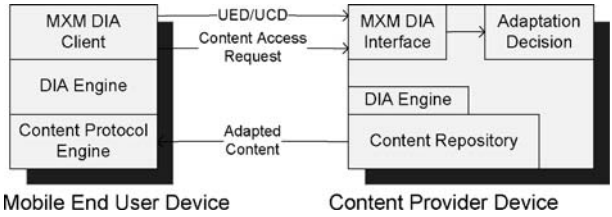

Fig. 4 Engines taking part in the content adaptation

service/content provider in a transparent way, i.e., independent of the actual transport protocol used (e.g., HTTP, RTSP).

However, still an open issue is the bi-directional signaling and aggregation of this context information in an interoperable and scalable way, specifically for the distributed, multi-step adaptation scenario as described in [12]. The procedure to render, store or transfer content in a portable device differs from the permanent device not only in the need for adaptation but in the lack of connectivity. For these cases, the mobile end user device must connect at least once a permanent end user device and transfer the content from the fixed to the portable device before rendering the content. MXM should provide in a future support to perform the transfer operation.

3.6 MXM for content distribution and broadcasting

DRM platforms offer different tools and APIs for distributing or broadcasting content. OpenSDRM media delivery server, or the Bluewhale Marlin Broadband Server are the equivalent modules in other architectures. Regarding DRM, MXM provides interoperability in nature, given that MPEG-21 REL has a profile for broadcasting which is specifically designed to work with other rights information standards in the broadcasting domain [3].

The standard content format allows the different parties in the value chain to operate uniformly on Digital Items, provided that the proper rights are acquired and the rules observed. Content may be stored on repositories and digital libraries (content provider devices), and it may be indexed, tagged, rated, streamed and traded. A standard set of APIs providing all users with an easy access to these technologies has the potential to maximise the flow and the value of content, and allow new services to spawn in the digital space.

3.7 MXM for graphic applications users

MXM is not only a middleware for DRM on MPEG-21, but it also provides the API for developing graphic applications based on MPEG-4. In addition with offering better compression for video and audio, MPEG-4 advancement over MPEG-1 and MPEG-2 also consists in support for a scene graph, graphics primitives and user interaction. Even though such features exist from its first version published in 1998, they are less known and deployed in current implementations of MPEG-4. One reason is the complexity of such representations. By proposing MXM for graphics as part of the Multimedia Framework Engine, MPEG aim is to facilitate the adoption

of such tools. Both the Media Framework Engine and the Rendering Engine provide this support.

The corresponding APIs was designed with respect to two requirements: to cover with a minimum set of methods the maximum number of applications and to keep completely invisible for the application designer the internal structure of encoders and decoders. This conducted to the standardization of four interfaces for 3D graphics: geometry, appearance, animation and texture/images which cover the entire graphics chain: the authoring tools are able to export in MPEG-4 format, the end-users applications such as media players or games are able to import MPEG-4 files. In addition, taking advantage of the DRM related APIs provided as well by MXM, distribution and trading of graphics assets are made possible in an easy manner, specially when combined with MPEG-7 metadata [10].

4 DRM in the MXM platform

DRM is not the only concern of MXM but at least a central one; therefore it provides the necessary elements to implement a complete DRM platform. DRM systems main elements include licensing mechanisms, license authorisers, customary content protection and accounting. As a novelty in this DRM system, there is a new element, the MVCO, an ontology of the media value chain which grants the conformance with the value chain model and the Intellectual Property rules.

4.1 Licensing

The terms and conditions by which a certain content can be served are expressed in what is called a license. Licenses are represented in the MPEG-21 REL language, which allows flexible licensing schemas even including special support for representing Creative Commons (licensing terms (the ORC profile is used to map the CC licenses within REL) and allows extensions of the language [21].

Content creators may issue pattern licenses, allowing them to declare some limits on the license terms that will finally be set for the end user (like giving a fixed price). Final product rights holders (either the same creators or third distributors) may issue end user licenses, constrained by the pattern licenses, defining the precise conditions and pricing—if any—for the user to render the Digital Item. This schema is similar to Coralt's one, with parallels with the Rights Token and DRM License they define.

Content creators can declare the pattern licenses for the content they create from the same application they use to create the content itself. A content creation application based on MXM also inherits the support to store the license in a remote license provider device (and eventually to revoke it).

The use of MPEG-21 REL allows expressing simple or complex licenses, specifying one or more rights between a set of predefined terms in a dictionary or specifying new extended rights. The permission to execute the rights can be conditioned to the prior satisfaction of a set of constraints including territorial limits and time ranges or limits in the number of executions. Licenses can be issued to either particular devices (every device is identified by a device ID) or to groups of devices called *domains* (in which case the identification is a domain ID).

Being a content item itself, MPEG-21 REL licenses can be individual XML files or can be bundled in the Digital Item packaged together with the rest of the content and metadata. So far, MXM has not engaged in the definition of the payment and license negotiation protocols, but it is not excluded that support for these technologies too may arise in the future.

MXM offers the API to create, edit and assess licenses in the corresponding REL engine, while the functionality to obtain and deliver licenses is given in the license protocol engine. Licenses can be issued to domains, in which case the domain engine API functionality is needed too.

4.2 Authorisation

The authorisation is the decision taken when a user wants to render a resource or make any other operation on a governed content. After the user has requested an action, appropriate licenses will be searched first in the content, then locally and lastly from a remote license provider if the other searches fail. Once a license is retrieved, the user request is evaluated against the current context and the license terms. The license will have granted some rights, executable only if certain conditions apply. Upon a positive authorisation, the MXM application will eventually retrieve the keys and algorithms to decrypt the content (in the case it was protected) with the IPMP tools offered by a DRM Tool provider device.

MXM does not specify how the authorisation algorithm should be implemented and it only points at the authorisation clause in the MPEG-21 Part 5 standard. MXM currently does not specify either whether authorisations should be performed locally in the usert's device or remotely in a secure environment. The first approach assumes that devices are trusted and not malicious, what can be granted by a tool certification mechanism. This tool certification mechanism could make an initial fingerprint of the software, and after an evaluation of the tool's behaviour, check regularly that the tool fingerprint is the same and the tool has not been manipulated.

Nothing prevents extending the licensing mechanism to distributors. Besides end user licenses, distribution licenses can be conceived, in which the action that is authorised is the mere act of distribution. While nothing prevents from chaining several links in the distribution chain with this schema, the MVCO Engine provides specifically methods to handle this requirement.

4.3 Content protection

Some content distribution platforms deliver content without protection, and this content has been often labeled as 'DRM-free'. The authors of this paper believe that this content may be called 'protection free', but not 'Digital Rights *Management* free', because *management* cannot be ignored in the process, especially when there are economic transactions in between.

Nevertheless, content protection is an optional feature offered by MXM. Content, either purchased or free, can be delivered encrypted or not. MXM does not specify a specific encryption algorithm: by means of the IPMP Engine it is possible to select an appropriate IPMP Tool to apply the required protection to content and the corresponding IPMP Tool Body (i.e. the hardware or software module implementing the specific IPMP function) will be used. For example, JPSEC encryption methods

could be provided for those Digital Items whose resources were images (Secure JPEG 2000 [2]) etc. The IPMP Tool protocol engine can be used in conjunction with the IPMP Engine to retrieve missing IPMP Tool Bodies from remote repositories and making them available on both the content creation and the content consumption side.

4.4 Accounting and value chain model rules observance

One of the technologies specified by the MPEG-21 standard refers to the notification of important events in the life cycle of the content, which can help the monitoring and accounting tasks a rights holder may need (either distributor, content creator or any other party which gives added value to the content or is involved in its distribution).

MPEG-21 Event Reporting provides standardized means for reportable events to be specified, detected and acted upon. An event report request is sent to the Event Report Engine whenever a user wants to be informed about certain event, and this central Event Report Engine will broadcast the event to all the concerned parties whenever an event has occurred (like a rendering, a license purchase, etc.).

Also and through the MVCO [13], it is possible to declare the kinds of Intellectual Property objects that are represented by Digital Items, their precedence relationships and the user roles involved, as well as to grant the respect to the Intellectual Property (IP) model along the Value Chain.

The Media Value Chain representation as an ontology represents both a conceptual and practical advantage. First, the IP-related concepts are clarified in an open, simple and precise manner. Open because it is public, simple as the model has been kept in its minimum agreed form, and precise because the exact semantic relations in the model are given as an OWL (Ontology Web Language) ontology. And second, it is practical because it allows the management of actual users, actions and IP Entities, where a clear distinction of the different IP Entities can be stated, and where users can know their IP roles respecting the particular IP Entities.

5 Conclusion

DRM lies in the center of MXM as it lied in the center of MPEG-21 when it was conceived in 2001. Ever since, there has not been a satisfactory solution in the media distribution playfield and abuses of the IP rights have not been overcome. Massive parallel distribution through P2P networks and other Internet channels has not ceased to exist and at most it has been palliated by tough legal prosecution.

Nowadays, Digital Rights Management is an unpopular term because it is associated to proprietary systems limiting the rights of users in the value chain. On the other hand, the rights management enabled by open, standard technologies made accessible by the MXM APIs may very well encourage an opposite reaction: an enabler of new media services, maximising the flow of content through the value chain rather than limiting it. May MPEG-21 framework adoption take off linked to a powerful and open DRM system, success chances of a more ordered trade of IP contents are not small.

The MXM platform provides a set of coherent technologies which empower the creation, distribution and use of digital media in a very harmonious and neatly defined way. Treating the value chain links with fairness and respect to each partyt's interests is a design feature of MXM. MXM platform does not lock to a particular device and its extensible and abstract definition allows MXM to be used in computers, mobile devices, set-top boxes and service platforms, at a quality level granted by the fact it has been adopted by standardization bodies. MXM source code is universally accessible and its use is royalty-free or at reasonable and non-discriminatory (RAND) conditions because the BSD licensing schema has been adopted.

A partial implementation in an advanced state is already available and can be found in [11], but the serious challenges are to arise later, when the real use of MXM will, as always, be the ultimate judge of the technology.

References

1. Chen X, Huang T (2007) Interoperability issues in DRM and DMP solutions. In: 2007 IEEE int conf on multimedia and expo, pp 907–910
2. Ebrahimi T, Grosbois R (2003) Secure JPEG 2000—JPSEC. Proc IEEE Int Conf Acoust Speech Signal Proc 4:716–719
3. Hogab K, Keunyoung L, Taehyun K, Wang X (2009) MPEG-21 and its interoperability with rights-information standards. IEEE Multimed 16(1):94–99
4. ISO/IEC 15938-12: Information technology—multimedia content description interface—part 12: Query format. International Standard, October 2008
5. ISO/IEC 23006-1: Information technology—MPEG-M (MPEG Extensible Middleware)—part 1: MXM architecture and technologies. Final Committee Draft, September 2009, accessible through [11]
6. ISO/IEC 23006-2: Information technology—MPEG-M (MPEG Extensible Middleware)—Part 2: MXM API. Final Committee Draft, September 2009, accessible through [11]
7. ISO/IEC 23006-3: Information technology—MPEG-M (MPEG Extensible Middleware)—Part 3: MXM conformance and reference software. Final Committee Draft, September 2009, accessible through [11]
8. ISO/IEC 29116-1: Information technology—supplemental media technologies—part 1: MPEG Extensible Middleware (MXM) protocols. Committee draft, March 2009, accessible through [11]
9. Koenen R, Lacy J, Mackay M, Mitchell S (2004) The long march to interoperable digital rights management. In: Proc. of the IEEE, vol 92, pp 883–897
10. Le Bonhomme B, Preda M, Prêteux F (2008) From MPEG-4 scene representation to MPEG-7 description. In: Multimedia semantics—the role of metadata
11. MPEG Extensible Middleware (2009) http://mxm.wg11.sc29.org. Accessed 22 Jul 2009
12. Ransburg M, Timmerer C, Hellwagner H (2008) Dynamic and distributed multimedia content adaptation based on the MPEG-21 multimedia framework. In: Multimedia semantics—the role of metadata
13. Rodriguez-Doncel V, Delgado J (2009) A media value chain ontology for MPEG-21. IEEE Multimed 16(4):44–51
14. Serrão C, Torres V, Delgado J, Dias M (2006) Interoperability mechanisms for registration and authentication on different open DRM platforms. IJCSNS 6(12):291–303
15. The Digital Media Manifesto (2003) In: Digital media project group. http://www.dmpf.org. Accessed 22 Jul 2009
16. Thomas-Kerr J, Burnett I, Ritz (2008) Format-independent rich media delivery using the bistream binding language. IEEE Trans Multimedia 10(3):514–522
17. Tokmakoff A, Nuttall F, Ji K (2005) MPEG-21 event reporting: enabling multimedia e-commerce. IEEE Multimed Mag 12(4):50–59
18. Torres V, Rodríguez E, Llorente S, Delgado, J (2004) Architecture and protocols for the protection and management of multimedia information. Lect Notes Comput Sci 3311:252–263

19. Torres V, Rodriguez E, Llorente S, Delgado J (2005) Use of standards for implementing a multimedia information protection and management system. In: 1st conf on automated production of cross media content for multi-channel distribution, p 197
20. Vetro A, Timmerer A (2005) Digital item adaptation: overview of standardization and research activities. IEEE Trans Multimedia 7(3):418–442
21. Wang X (2004) MPEG-21 rights expression language: enabling interoperable digital rights management. IEEE Multimed 11(4):84–87

Víctor Rodríguez Doncel is studying toward a PhD in Semantic Web and Intellectual Property at the Universitat Politècnica de Catalunya. His research interests include secure e-commerce, ontology modeling, and artificial intelligence. Rodriguez Doncel has a degree in telecommunication engineering from the Universidad de Valladolid, Spain.

Jaime Delgado is a professor in the Computer Architecture Department of the Universitat Politècnica de Catalunya, and is head and founder of DMAG. His research interests include electronic commerce, digital rights management, metadata interoperability, multimedia content search, security in social networks, and distributed multimedia applications. Delgado has a PhD in telecommunication engineering from Universitat Politècnica de Catalunya.

Filippo Chiariglione is co-founder and CEO of SmartRM Inc, a start-up company providing social networking-oriented applications and services to protect the confidentiality of users' audiovisual content and documents. Before this position, he was a Technology Specialist with CEDEO.net in Turin, Italy, a consulting company in digital media; Senior Engineer with the Multimedia Group, EMI - Central Research Labs, London, and Visiting Student with the Internet Media Group at Microsoft Research Asia, Beijing. Filippo received his PhD in 2009 from the University of Florence for research on technologies to manage rights and value of digital media, and graduated in Computer Science Engineering from Politecnico di Torino in 2002. He has contributed to several ISO MPEG standards for which he received ISO/IEC certificates and has published a number of papers, book chapters and tutorials in the area of his research.

Marius Preda received an Engineer Degree in Electronics from University POLITEHNICA (Bucharest, Romania) in 1998, and a Ph.D. Degree in Mathematics and Informatics from University Paris V - René Descartes (Paris, France) in 2001. He is currently Associate Professor at Institut Telecom. His research interests include generic virtual character definition and animation, rendering, low bitrate compression and transmission of animation, and multimedia composition and standardization. He is the chairman of the 3D graphics subgroup of ISO's MPEG (Moving Picture Experts Group) and is involved in several research projects at the institutional, national and European levels.

Christian Timmerer is an assistant professor in the Department of Information Technology (ITEC), Multimedia Communication Group (MMC), Klagenfurt University, Austria. His research interests include the transport of multimedia content, multimedia adaptation in constrained and streaming environments, distributed multimedia adaptation, and Quality of Service / Quality of Experience. He was the general chair of WIAMIS 2008 and EUMOB 2009 and has participated in several EC-funded projects, notably DANAE, ENTHRONE, P2P-Next, and ALICANTE. He is an associate editor for IEEE Computer Science Computing Now, a new online front end to all IEEE CS magazine which transitions the printed media to virtual adopting and applying Web 2.0 principles in practice. He also participated in ISO/MPEG work for several years, notably in the area of MPEG-21, MPEG-M, and MPEG-V. He received his PhD in 2006 from the Klagenfurt University. Publications and MPEG contributions can be found under http://research.timmerer.com, follow him on http://www.twitter.com/timse7, and subscribe to his blog http://blog.timmerer.com.