

Design Principles and Issues of Rights Expression Languages for Digital Rights Management

Xin Wang

Contentguard, Inc., 222 N. Sepulveda Blvd., Suite 1400, El Segundo, CA, USA 90245

ABSTRACT

Digital rights management (DRM) provides a unified approach to specifying, interpreting, enforcing and managing digital rights throughout the entire life cycle of digital assets. Using a declarative rights expression language (REL) for specifying rights and conditions in the form of licenses, as opposite to some other approaches (such as data structures and imperative languages), has been considered and adopted as a superior technology for implementing effective, interoperable and scalable DRM systems. This paper discusses some principles and issues for designing RELs, based on the experiences of developing a family of REL's (DPRL, XrML 1.x, XrML 2.0 and MPEG REL). It starts with an overview of a family tree of the past and current REL's, and their development history, followed by an analysis of their data models and a comparison with access-control oriented models. It then presents a number of primary design principles such as syntactic and semantic un-ambiguity, system interoperability, expressiveness in supporting business models and future extensibility, and discusses a number of key design issues such as maintaining stateful information, multi-tier issuance of rights, meta rights, identification of individual and aggregate objects, late-binding of to-be-identified entities, as well as some advanced ones on revocation and delegation of rights. The paper concludes with some remarks on REL profiling and extension for specific application domains.

Keywords: Rights Expression Languages, Digital Rights Management, RELs, DRM.

1. INTRODUCTION

Digital rights management (DRM) refers to the collection of hardware, software, services, and technologies for persistently governing authorized distribution and use of content and services according to their associated rights and managing consequences of that distribution and use throughout their entire life cycle or workflow. For DRM, content can be in the form of audio, video, text, image, software or others, and a service can be any local or remote, centralized or distributed resource that provides a set of functionality to its clients. Central to DRM is a unified approach to specifying, interpreting, enforcing and managing digital rights throughout the entire life cycle of digital assets. Security techniques such as identification, authentication, encryption, digital signature, and watermarking are employed to serve the purpose of honoring and enforcing rights in DRM systems^{12,13,21}. For instance, if one has no rights to use a piece of content, then he/she should not be provided with a key to decrypt encrypted content.

One of the key technologies in digital rights management (DRM) is a rights expression language (REL). In order to develop effective and efficient DRM systems, the capability of specifying and communicating rights information among the participants is certainly required at each step of the lifecycle or in the workflow. For example, backward along the supply-distribution-consumption value chain, content users need to know what rights are associated with content and granted to them, content distributors and retailers need not only to communicate the rights that are available for consuming the content but also to understand the rights that pertain for distributing the content, and content providers in the upstream of the value chain need to ensure that both usage and distribution rights are granted precisely as intended for each participant in the end-to-end value chain. Clearly, these rights can be simple or very complex. For example, a user may obtain the rights for unlimited play for a music file, and a corporate document may have the usage right restricted to certain management levels and business divisions. Rights expressions get more complex when one tries to model the use and distribution of content in the physical world. For example, specifying the rights that govern the lending of a digital book or the giving-away of an article in an electronic magazine could be fairly complex.

Thus, a common REL that can be shared among the participants in the workflow and lifecycle is essential. Not only from an obvious interoperability point of view, but more so to comprehend that rights will be manipulated and changed during the workflow and lifecycle as content moves from the creator, aggregator, distributor, retailer, and finally to

consumer, and to comprehend system issues such as security and trust needed to preserve the authenticity, integrity, confidentiality and trustworthiness of rights expressions.

In the last few years, there have been a large amount of efforts devoted to developing open REL standards. The most notable ones are the REL from the Moving Picture Expert Group (MPEG)^{10,16,34}, which is application neutral and of general purposes, and the REL from the Open Mobile Alliance (OMA)²², which is domain specific to the mobile applications. These two standard RELs are all declarative languages defined using the W3C XML Schema³² and DTD²⁹, respectively. Their existence has demonstrated that using a declarative language for specifying rights and conditions a DRM technology superior to some other approaches (such as using data structures and imperative languages) for implementing effective, interoperable and scalable DRM systems.

This paper discusses some principles and issues for designing an REL, based on the experiences of developing a family of REL's (DPRL, XrML 1.x, XrML 2.0 and MPEG REL). It starts with an overview of a family of the past and current REL's, and their development history. It then presents an analysis of their data models to illustrate how organize rights information at an abstract level, followed by a comparison with access-control oriented data models and a discussion on some commonalities and differences between DRM and access control, with a goal to show why DRM has a large application scope and deals with more technological challenges than access control. After that, the paper provides a number of primary design principles such as syntactic and semantic un-ambiguity, system interoperability, expressiveness in supporting business models and future extensibility, and discusses a number of key design issues such as maintaining stateful information, multi-tier issuance of rights, meta rights, identification of individual and aggregate objects, late-binding of to-be-identified entities, as well as some advanced ones on revocation and delegation of rights. The paper concludes with some remarks on REL profiling and extension for specific application domains.

2. FAMILY OF REL'S

The development of rights expression languages has already had an over-ten-year history. The following figure shows a development time-line for a number of past and current REL's (see also references^{8,37}).

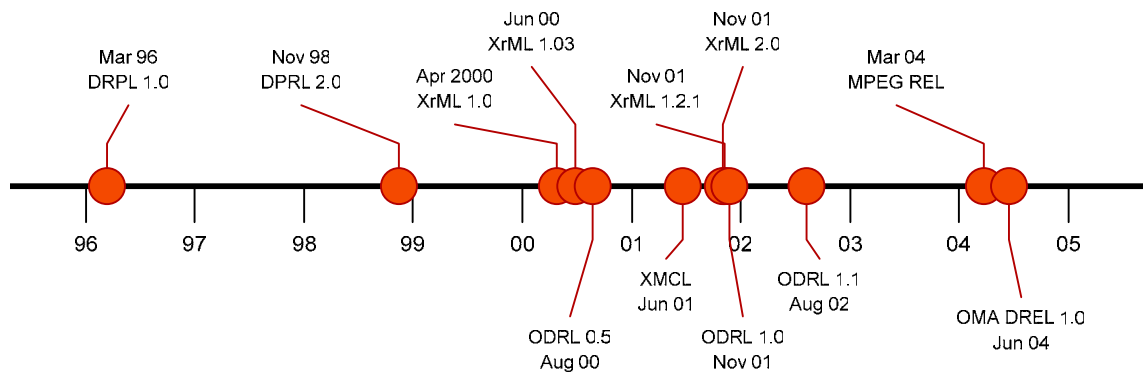


Figure 1: Development history of rights expression languages.

The first development of a rights expression language was started by Mark Stefik in 1994 at Xerox's Palo Alto Research Center (PARC). The result of it was a computer-interpretable language, called the Digital Property Rights Language (DPRL), for describing rights, conditions, and fees for using digital works. The first version of DPRL, v1.0, originally written in the programming language LISP, was released in March 1996, and the second version, v2.0, defined using the XML DTD²⁹, in November 1998³⁶, which enabled specification of additional rights, fees, terms and conditions in a uniform, flexible and extensible manner. From the very beginning, DPRL was intended to support commerce in digital works, that is, publishing and selling electronic books, digital movies, digital music, interactive games, computer software and other creations distributed in digital form. DPRL was also intended to support specification of access and use controls for secure digital documents in cases where financial exchange is not part of the terms of use. As the pioneer REL, DPRL laid out the fundamental concepts and approach to all other later-developed RELs. Moreover, it demonstrated the advantages of the declarative language approach over other approaches for specifying rights information, including ones using low-level system specific data structures (e.g., bits) and imperative

languages (e.g., executables), in terms of precise semantics, implementation independence, flexibility, and extensibility. It also intended to become a standard to enable interoperability, as “the use of a standard language for usage rights on digital property ensures that trusted systems can exchange digital works and interoperate”³⁶.

In 1999, as ContentGuard, Inc. prepared to be sponged off from Xerox to continue the development of an open standard REL, DPRL was renamed to the eXtensible rights Markup Language (XrML), to reflect the adoption of the W3C XML standard²⁹ as a syntactic foundation for an extensible mark-up language to specify rights information. The XrML’s use of XML as the meta-language for defining an REL sets the direction for essentially all later RELs. The first version of XrML, v1.0, was introduced in April 2000, when ContentGuard became an independent, spin-off company from Xerox. Soon after that, a next public version, XrML v1.03, was released and submitted to the Electronic Book Exchange (EBX) working group (which was later merged with the Open eBook Forum (OeBF)²¹). It was after the open discussions at the EBX about using a language to specify rights information, other RELs started to surface; they include a draft version of the Open Digital Rights Language (ODRL)²⁰, v0.5, released in August 2000, and an initial draft of RealNetworks’ Extensible Media Commerce Language (XMCL)²³, released in June 2001. In November 2001, the XrML 1.x series reached their last version 1.2.1, which is the REL currently used in the Microsoft Rights Management Services (RMS) related products¹⁴.

The development of XrML had adopted the design philosophy of DPRL to enable trusted systems to interoperate for end-to-end DRM systems, and strived to serve as the basis in defining an industry standard rights language. This effort led to the release of XrML v2.0 in November 2001^{4,3534}, which was redefined using the W3C XML namespaces and schemas^{30,32}, with support for more business models and enhanced security, flexibility and extensibility. Its release was in time for responding to the MPEG’s Call for Proposals for developing the MPEG REL¹⁵. After a careful evaluation process at the MPEG meeting in December 2001, XrML 2.0 was selected as the core architecture and base technology over the other submissions including especially ODRL v1.0²⁰, for its precise semantics, flexibility to support more business models and sound extensibility architecture.

After more than two years of work by representatives from technology companies, consumer electronics companies, content owners, and creators that have taken part in an open development process, MPEG published its REL as an International Standard¹⁰ in March 2004, together with its sister specification, MPEG Rights Data Dictionary (RDD)¹¹, as part of the MPEG-21 standard specification suite^{1,2}. The MPEG REL is a single rights expression language that can be used across all media types, platforms, formats, resources, products and services to facilitate interoperability among DRM systems. It has an application and domain agnostic structure. It is comprehensive to express wide variety of business models, applicable to all phases of the life cycle of digital works, extensible to allow adaptability and minimize future cost of change, and flexible to be profiled for ease of implementation in specific DRM application domains.

The MPEG REL has started to receive a lot of attention. It had been selected over ODRL 1.1²⁰ (dated in August 2002) by the Open eBook Forum (OeBF)’s Rights and Rules Working Group²¹ to define an REL standard for OeBF. It is also under consideration for adoption by a number of other standards bodies due to its nature of being domain agnostic. For instance, the OASIS Web Services Security Technical Committee¹⁹ has approved the use of the MPEG REL with respect to the WSS specification for building secure Web services to implement message level integrity and confidentiality. In addition, there is now an on-going liaison between the Learning Technologies Sub-committee of the IEEE⁹ and MPEG, specifically on adopting the MPEG REL for the delivery of distance teaching and learning. Most recently, the Digital Media Project⁷ has also decided to use a profile of the MPEG REL in its Interoperable DRM Platform (IDP) specification.

While MPEG was developing its REL, the Open Mobile Alliance (OMA)²² was also working on its domain-specific REL. OMA released its first version, v1.0, in June 2004, which was based on a profile of the ODRL version 1.1²⁰, and designed specifically for mobile DRM applications. Its second version, v2.0, is now under development.

3. DATA MODELS OF REL’S

In developing an REL, it is very critical to utilize a number of abstractions for organizing rights related information. A data model of an REL is an abstraction in identifying essential pieces of rights information and their mutual relationship

as expressed in the REL. This section starts with a list of essential elements that are common in the existing RELs, shows data models of the DPRL/XrML 1.x and XrML 2.0/MPEG REL, and contrasts these models with the ones in (discretionary) access control, and discusses some commonalities and differences between rights expressed in RELs and policies in access control.

3.1 Data Model Elements

Consider an example, where a song, “When the Thistle Blooms”, is distributed by a label, “PDQ Records”, to the owner of an MP3 player, named “Alice”, with the right allowing her to play the song for 3 weeks. The rights information that an REL needs to express is that “Alice is granted by PDQ Records with the right to play ‘When the Thistle Blooms’ for 3 weeks”.

Essentially, there are five data model elements in this rights information, as shown in Figure 2. In the XrML 2.0/MPEG REL terminology, they are, as shown in Figure 2, “principal”, “right”, “resource”, “condition”, and “issuer”. An REL should at least be able to express a right granted by an issuer to a principal for a resource and any condition under which the right can be exercised. In the example above, “PDQ Records” is an issuer, “Alice” is a principal, “When the Thistle Blooms” is a resource, “play” is a right, and “for 3 weeks” is a condition.

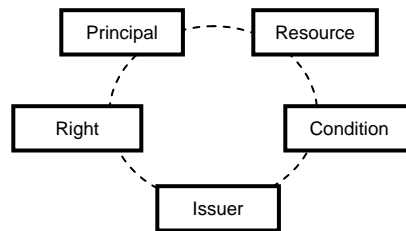


Figure 2: REL data model elements.

3.1.1. Principal

A principal is a “subject” or “party” to whom a right is granted. A principal denotes the party or subject that it identifies by information unique to that party. Usefully, this is information that has some associated authentication mechanism by which the principal can prove its identity. For example, the MPEG REL supports the concept of a key holder, meaning someone identified as possessing a secret key, such as the private key of a public/private key pair. The principal element should be extensible, so that different mechanisms for identifying principals can be defined.

3.1.2. Right

A right is the “verb” that a principal can be granted to exercise against some resource under some condition. Typically, a right specifies an act (or activity) or a class of acts that a principal may perform on or using the associated resource. For instance, the MPEG REL provides a set of commonly used, specific rights, such as play, print and adapt, as well as rights relating to other rights, such as obtain, issue, and revoke. This right element should be extensible, so that different rights for different applications can be defined.

3.1.3. Resource

A resource is the “object” to which a principal can be granted a right. A resource can be a digital work (such as an eBook, an audio or video file, or an image), a service (such as an email service, or B2B transaction service), or even a piece of information that can be owned by a principal (such as a name, an email address, a role, or any other property or attribute). A resource can also be a rights expression itself. For example, the MPEG REL provides mechanisms to encapsulate the information necessary to identify a particular resource or a collection of resources with some common characteristics. This resource element should be extensible, so that different mechanisms for identifying resources can be defined.

3.1.4. Condition

A condition specifies the terms, conditions, and obligations under which rights can be exercised. For example, a simple condition is a time interval within which a right can be exercised. A slightly more complicated condition may require

the existence of a valid, prerequisite right that has been issued by some trusted entity. Using this mechanism, the eligibility to exercise one right can become dependent on the eligibility to exercise other rights. Moreover, a condition can be the conjunction of several other conditions. For instance, the MPEG REL defines conditions appropriate to using digital works (for instance, fee, destination, and territory). This condition element should be extensible, so that different conditions for different applications can be defined.

3.1.5. Issuer

An issuer identifies a principal who issues rights. The issuer can also supply a digital signature signed by the principal to signify that the principal does indeed bestow the rights issued, and to facilitate reliable establishment of trustworthiness of the rights information by others.

3.2 Data Model of DPRL and XrML 1.x

In DPRL and XrML 1.x, the data model elements and their inter-relationships are organized as shown in Figure 3.

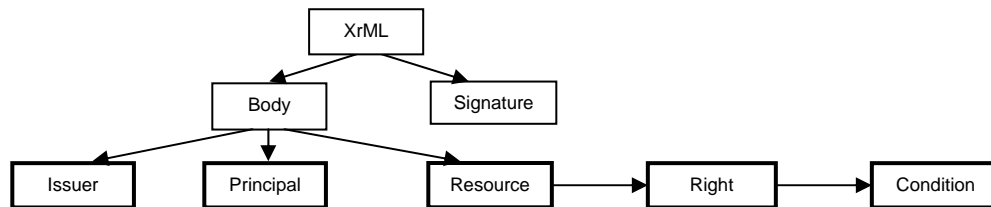


Figure 3: Data model of DPRL and XrML 1.x.

In this model, a rights expression is a DPRL/XrML 1.x document that consists of a body and a signature that ensures the information integrity of the body. The body specifies an issuer, a principal (called “issuedPrincipal”) and a resource (called “work”), and makes a statement that the issuer grants some rights (to be specified within the resource) to the principal over the resource. The resource contains a list of rights granted over the resource, and each right in the list may have a list of conditions attached to it, under which the right can be exercised.

According to this model, to interpret if a principal can exercise a right over a resource against a rights expression, the interpreter needs to perform the following tasks:

- to validate the signature in the expression to ensure the information in the body is authentic,
- to verify if the issuer is trusted
- to check if the principal matches with the one specified in the body
- to check if the resource matches with the one specified in the body
- if the resource matches, check if the right is in the list of rights associated with the resource, and
- if the right is in the list, check if there are any conditions attached to the right, and if there are, check if each of the conditions is satisfied.

When all the tasks are performed successfully, then the request to exercise the right is authorized, and otherwise denied.

Notably, DPRL and XrML v1.x contain elements for specifying generic rights so that new rights can be defined, and a number of pre-defined rights, such render (play, print, view, export), transport (copy, transfer, loan), derivative work (edit, extract, embed), file management (backup, restore, verify, folder, directory, delete) and configuration (install, uninstall). They also have elements for terms and conditions, such as those related to time (from, until, interval, metered), access control (principal, security level), fee (flat, per-use, metered, ticket), territory (location, domain) and obligation (tracking, watermarking).

3.3 Data Model of XrML 2.0 and MPEG REL

XrML 2.0 and MPEG REL adopt a data model shown in Figure 4 that is different from their predecessors. In this data model, a rights expression is a license that contains one or more grants and an issuer that issues the rights specified within the grants, and each of the grants contains a principal (optional), a right, a resource (optional), and a condition (optional). The issuer can also contain a digital signature of the license signed by the principal to signify that the principal does indeed bestow the grants contained in the license, and to facilitate reliable establishment of

trustworthiness of the license by others. Semantically, a license is a collection of grants issued by an issuer, and each grant makes a statement that the principal (or anyone if the element is omitted) has the right over the resource under the condition (or under no condition if this element is omitted).

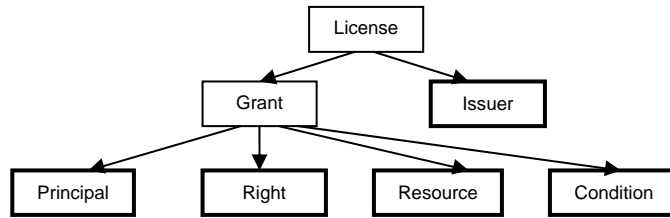


Figure 4: Data model of XrML 2.0 and MPEG REL.

According to this model, to interpret if a principal can exercise a right over a resource against a license, the interpreter needs to perform the following tasks:

- to verify if the issuer is trusted, and to validate the signature, if any, associated with the issuer to ensure that the related information in the license is authentic,
- for each grant in the license,
 - if there is a principal in the grant, check if it matches with the requesting principal
 - check if the right in the grant is the requested right
 - check if the resource in the grant matches with the requested resource (and if the resource is omitted, then the requested resource should also be omitted)
 - if there is a condition in the grant, check if the condition is satisfied.

When the issuer verification is successful, and there is a grant for which all the checks are successful, then the request to exercise the right is authorized, and otherwise denied.

3.4 Data Models of Access Control

Access control²⁶ is the process of mediating every request to data and services maintained by a system and determining whether the request should be granted or denied. In the access control terminology, a principal is called a subject, and a resource an object. It is interesting to compare the REL data models with those used to organize information in access control policies.

In access control, a conceptual model that specifies the rights that a principal possesses for a resource is represented by an access control matrix. In this matrix, there is a row for each principal and a column for each resource. Each entry of the matrix specifies a list of rights granted to the principal in the row to the resource in the column. Thus, the access matrix can be captured as the data model shown in Figure 5.

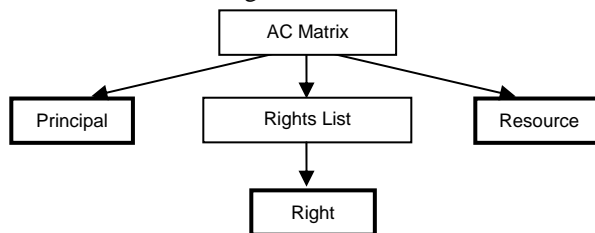


Figure 5: Data model of the access control matrix.

In this model, to interpret if a principal can exercise a right over a resource against an access matrix, the interpreter needs to perform the following tasks:

- check if there is a row for the principal
- check if there is a column for the resource
- check if the right is in the list of rights corresponding to the principal in the row and the resource in the column.

When all the tasks are successful, then the request is authorized, and otherwise denied. Clearly, this model does not consider the condition and issuer elements, and its application scope is limited.

In a large system, the access control matrix will be enormous in size in terms of the numbers of its principals and resources the system has, and most of its entries are likely to be empty. Accordingly, the access control matrix is very rarely implemented as a matrix. A popular approach to implementing the access control matrix is by means of access control lists (ACLs). Each resource is associated with an ACL, indicating for each principal in the system the rights the principal is granted to exercise on the resource. This approach corresponds to storing the matrix by columns. A dual approach to ACLs is the one using capabilities. In the capability approach, each principal is associated with a list (called the capability list) that indicates, for each resource in the system, which rights the principal is granted to exercise on the resource. The data models for these two approaches are shown in Figure 6 (for ACLs) and Figure 7 (for capabilities).

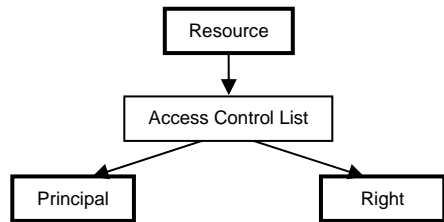


Figure 6: Data model of the access control list.

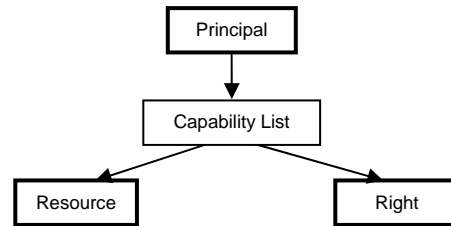


Figure 7: Data model of the capabilities.

In these models, interpreting if a principal can exercise a right over a resource against an access matrix is also straightforward. For instance, the interpreter needs to perform the following tasks for ACLs:

- check if there is an ACL for the requested resource
- if there is, check if it has an entry matching the requesting principal
- if it matches, check if the requested right is in the list of rights corresponding to the principal.

When all the tasks are successful, then the request is authorized, and otherwise denied.

3.5 Some Commonalities and Differences between DRM and Access Control

From the authorization point of view, DRM shares an ultimate goal with access control (AC); that is, both of them want to provide mechanisms to decide whether or not a principal is allowed to exercise a certain right over a resource. In DRM, rights granted are specified in the forms of an expression in some REL, whereas, in AC, rights are defined in some policies such as access control lists and capabilities (whose actual implementations can be using low-level system features like bits). In making authorization decisions, almost all DRM and AC systems adopt a closed-world assumption; that is, to a DRM or AC system, the only rights a principal has are those explicitly granted (by their trusted issuers) to it and accessible by the system, and only rights explicitly granted (by their trusted issuers) are allowed to exercise by the system. This closed world assumption is in contrast to the open world assumption that allows any principal to exercise any right unless explicitly specified otherwise. Nevertheless, both DRM and AC rely on security mechanisms to identify, and authenticate relevant entities such principals/subjects and resources/objects, in some more or less sophisticated manners.

However, there are a number of aspects that set DRM different from AC, three of which are discussed here. One is concerned with the ownership of resources and their associated rights. DRM deals with, in most cases, situations (see the example in Section 3.1) where a principal (e.g., user “Alice”) wants to exercise rights (e.g., “play”) over resources (e.g., song “When the Thistle Blooms”) created by others, and the rights are granted by the rights owners of the resources or their delegates (e.g., label “PDQ Records”) to the principal. In contrast, AC aims at regulating access to resources within a security domain. As an important concept in security, a domain is defined in as “an environment or context that is defined by a security policy, security model, or security architecture to include a set of system resources and the set of system entities that have the right to access the resources”²⁷. AC itself does not address where system resources come from originally and if different policies should be applied for foreign resources. This results in that, once these resources get into a domain (e.g., via file downloading from a file server, a peer-to-peer system or a portable media), these resources are under the control of the AC policy defined by the domain administrator, rather than the rights defined by rights owners of the resources. Usually, a domain AC policy grants all the rights to the owner of the

domain over any resources within. This practice has been the root for causing casual rights violation and deliberated content piracy. It is this problem that leads to a persistent rights-resource association requirement to an REL and DRM as whole; that is, rights have to be associated with their resources in a persistent manner throughout the life-cycle of the resources, even when the resources are transported from one (security) domain into another. In this way, a DRM system can then effectively enforce rights according to authentic rights granted by the original rights owner. Therefore, it is fair to state that DRM deals with inter-domain management of rights, while AC mainly focuses on intra-domain management of rights.

The second aspect, related to the first one, is about trust management of all entities involved in authorization, especially authorization policies. Generally speaking, the problem of authorization can be divided into two related subproblems: specification and interpretation. Specification is about specifying or expressing authorization policies in some machine-readable form, whereas interpretation refers to evaluating an authorization request against a given set of authorization policies and determining if the request should be granted or denied. The separation of specification from interpretation makes it possible that an entity (issuer) who issues a policy is different from an entity (interpreter) who interprets the policy. In DRM, it is very important to establish a trust relationship between the issuer and the interpreter, so that rights granted by trusted issuers are rightfully processed by trusted interpreters. In AC, this has not been an outstanding issue, as a policy in a domain is usually defined by its domain administrator and interpreted by a reference monitor which is trusted within the domain²⁶. Because of this, most of AC policies do not carry information for establishing such trust relationship. But, to ensure that rights expressions are issued by trustworthy issuers, an REL must provide elements, such as license issuer and signature in the MPEG REL, to enable this kind of trust management.

The third aspect is on managing multi-tier rights along the distribution value chain of resources. In DRM applications, resources and multimedia content in particular are distributed and transferred from content owners, to distributors and retailers, then to end-users, and even possibly further shared among end-users within an authorized domain. Rights need to be specified for all the participants throughout the distribution value chain in the form of multi-tier rights. For instance, a content owner may want to grant a distributor a right to issue usage rights to end-users. The right the distributor gets is then a two-tier right, as it is about not just the distributor's right to issue but also what rights the distributor can issue to end-users. AC, on the other hand, focuses on specifying and interpreting policies for each single access point, and rarely considers how an access point may affect another access point in the value chain. Thus, it is important that an REL should support a type of rights, called meta rights, which can be used to deal with other rights. The right to issue rights is an example of meta rights. Others include the rights to obtain and revoke rights. With meta-rights, multi-tier rights can be specified with meta rights over other rights, which could include other meta-rights. If an REL is capable of expressing these meta and multi-tier rights, the REL is then capable of supporting many practical business models.

4. PRINCIPLES

As an REL is a vital component in DRM to specify rights and conditions for authorized distribution and use of any content, resources and services, there are a number of design principles that developers should follow in the aspects of its semantics, expressiveness, extensibility, flexibility, and support for interoperability.

4.1 Unambiguous Semantics

An REL should have well-defined syntax and semantics that can be used to specify rights unambiguously. In particular, it should have an independent semantics that is separated from its implementation in system-specific mechanism. This is fundamentally important because the language is a vehicle to communicate the meanings of expressions written in the language to all the participants in DRM systems.

An example that often causes semantics ambiguity is a resource that represents a content collection, such as a music play list or a music subscription list. When granting a right to play that resource, what is the meaning of the right? Intuitively, in the play list case, it makes sense to play the entire list as a whole, i.e., to play all the songs, one by one continuously, in the list, whereas, in this subscription list, it is meaningful to play any song in the list, but on the individual base, not all the songs in the entire list altogether. Thus, if the specification of the right to play a resource of a collection type uses a same mechanism in these two cases (say, "play" a resource identified by a play list identifier "playlist-id:01234" or a subscription list identifier "subscriptionlist-id:56789"), then the semantics of the language is

dependent on the meaning of each identifier and hence potentially ambiguous. To resolve the issue in this case, it is better to assign the semantics of playing a resource as playing the entire resource (like the play list), and to specify the right for a subscription list as the right to play every resource in the subscription list, instead of playing the subscription list resource. The variable mechanism provided in the MPEG REL¹⁰ supports the latter case; see^{5,6} for more examples.

In order to avoid ambiguity and to ensure being well-defined for every rights expression that can possibly be specified in the language, it is ideal that an REL has a formal semantics, like any other mature computer language does. This way, the meaning of each rights expression in the language can be precisely and uniquely determined, to reflect the intent of the issuer of the expression and to convey precisely one and only one interpretation what have been granted in the expression. To this end, the MPEG REL¹⁰ provides an authorization model to determine if an authorization or access control request can be granted according to a set of rights expressions in the language.

4.2 Expressiveness in Supporting Business Models

As an important DRM technology to enable electronic commerce and enterprise management, an REL must be rich enough in its expressiveness to support a wide variety of, existing and new, usage models in the end-to-end distribution value chain for all types of digital content, resources and services in closed and centralized as well as open and distributed environments.

This means that an REL should be comprehensive in expressing simple and complex rights expressions in any stage in a workflow, lifecycle or business model. The comprehensiveness includes:

- capability to express multi-tiered content distribution scenarios. For example, transfer, transform and delegation of rights down the distribution chain where the transactions may be from publishers to distributors, distributors to consumers, or consumers to consumers;
- capability to associate different rights and conditions to parts of a composite multimedia digital content;
- capability to associate rights and conditions and to satisfy the requirements for the entire life cycle of digital content from creation and aggregation through distribution to consumption and usage tracking;
- integration of multiplicity of rights and conditions as a group of which can be processed as a single consolidated entity for a business purpose; and
- management of rights processing such as issuing, obtaining, delegation, revocation, and expiration of rights.

4.3 System Interoperability

A complete, functional DRM system requires interoperability among all involved components possibly from different vendors. An REL shall support those elements that are required for components to interoperate within the context of an end-to-end DRM system. Those elements include identification and authentication of entities, integrity, confidentiality and trustworthiness of resources and even rights expressions themselves, and accounting and tracking of usage and distribution of resources and exercising of rights.

A good example to highlight this system interoperability principle is specification and tracking of state information related to exercising rights. For instance, if one is allowed to play a video clip five times, then merely specifying this play right constrained with the five-time condition is not enough to enable interoperability among different DRM systems that interpret and enforce the right. This is because, if the user plays the video three times using one DRM system, and wants to play it again but using another DRM system, the specification does not contain enough information to help the two systems to share the state information that the user has played three times, and the two systems wouldn't be able to interoperate in a consistent way that honors the right in a way that does not require system-dependent coding of the state information. The MPEG REL¹⁰ provides elements for specifying services so that state information can be tracked and retrieved from a service specified in a rights expression.

4.4 Extensibility

An REL has to be open-ended to allow future and domain-specific extensions, especially in terms of its data model elements, principal, right, resource and condition. Therefore, it should have simplicity of extension development, employ standard methods for extensions, and demonstrate actual extensions to the language itself.

In this regard, the MPEG REL is designed to be extensible and is itself specified in extensions. Its syntax is described and defined using the XML Namespace and Schema technologies defined by W3C^{30,32}. The extensive use of XML Schema enables the REL to offer a significant amount of extensibility and customizability without requiring actual changes to its core. Indeed, the MPEG REL itself makes use of this extensibility internally; its standard and multimedia extensions are defined using the extension mechanisms employed by the language.

4.5 Flexibility

An REL has to be flexible in using its functional features to meet different requirements from different applications ranging from the entertainment industry to enterprises and individuals. For an XML-based REL like the MPEG REL, this requires that mandatory and optional elements and their types, and mandatory and optional attributes and their values are clearly-defined, and that all possible expressions resulting from those elements and attributes are all well-defined in terms of their syntax and semantics.

5. ISSUES

Following the design principles above, there are many design issues and considerations in developing a precise, functional and comprehensive REL that enable interoperability in DRM systems. This section only highlights a few of them, due to the space limitation of this paper.

5.1 Entity Identification and Authentication

This issue is about specification of information needed to assist identification and authentication of individual as well as aggregate entities such as principals and resources in an REL. For instance, the MPEG REL provides the `keyHolder` element for an individual principal that possesses a cryptographic key, and the `allPrincipals` element for an aggregate principal that is identified by conjunction of possibly more than one other principals.

5.2 State-Information Specification and Tracking

The system interoperability principle was illustrated above with tracking of stateful information. Specifying what state information needed to be tracked and how it is tracked is a challenge. The MPEG REL provides a service reference mechanism for this purpose which uses web services standards like UUDI and WSDL.

5.3 Multi-Tier Rights Issuance

To enable multi-tier applications, not only end-user rights but also distributor rights need to be specified. For instance, it should be possible to specify distributor's rights to issue end-user's rights. Rights of this nature, that is, rights about other rights, are called meta-rights. The MPEG REL provides meta-rights like `issue`, `obtain` and `revoke`.

5.4 Late Binding

In many multi-tier distribution cases, it is quite often that some entities (such as a principal or resource) wouldn't get resolved until a late stage in the distribution chain. For instance, when a content owner issues rights to a distributor allowing it to distribute some content to end-users in a certain group, the content owner is unable to resolve to which actual end-user the distributor is to issue an end-user rights expression. It is at the time the distributor wants to (exercise its right to) issue rights to an actual end-user that the distributor can resolve the identity of that user. Hence, in the rights expression that the content owner issues to the distributor, there should be elements that enable this type of late-binding of entities. The MPEG REL supports declaration and reference of variables that allow variable values to be filled in at late stages in the distribution value chain.

5.5 Integrity and Trust of Rights

In order to support trust management at the rights expression level, an REL should provide mechanisms that enable attestation of the integrity of rights expressions, so that one can determine if the expression is trusted or not. This is very important when there is a chain of issuance of rights along the multi-tier distribution value chain, where one needs to verify if an upper-stream issuer has proper rights to issue rights for a down-stream issuer or end-user. The MPEG REL supports it with the `license signature` element.

5.6 Rights Delegation and Revocation

DRM systems are dynamical systems, in terms of principals, rights, resources, and conditions involved, as they may change over times. One user may want to allow another user or a device to exercise a right on his behalf, but does not allow passing the same right further to others. This raises the issue of rights delegation and control along a delegation path. In other cases, a former user may not be with a DRM system any more (due to reasons like leaving a company, canceling a subscription, or changing identity), and any rights issued to that user need to be revoked (even before the rights are expired). Developing effective and efficient mechanisms for rights delegation and revocation has been an interesting technical problem. The MPEG REL provides several elements like revocable resource, revoke right, and delegationControl and revocationFreshness elements for supporting these features.

6. CONCLUDING REMARKS

An REL is a key component of DRM systems. Developing a functional and practical REL needs to be guided by a number of design principles, and resolve many technical issues. The two existing standard RELs, one from MPEG and other from OMA, represent the outcomes of two approaches in system architecture design: the former being top-down, and the latter bottom-up. Consequently, the MPEG REL is a general-purpose language, whereas the OMA REL is a domain specific language. In order to meet requirements from a specific application domain, the MPEG REL may need to be profiled and extended (at its extension points). But, in a long run, in terms of enabling the DRM interoperability, the MPEG REL certainly has its advantage in the potential to serve as a common REL to facilitate interoperability among different DRM systems, across different application domains, and along the distribution value chain.

REFERENCES

1. J. Bormans, J. Gelissen, and A. Perkis. "MPEG-21: The 21st Century Multimedia Framework", *IEEE Signal Processing Magazine*, Volume 20, Issue 2, Mar 2003. Pages 53- 62.
2. I. Burnett, R. Van de Walle, K. Hill, J. Bormans, and F. Pereira. "MPEG-21: Goals and Achievements". *IEEE Multimedia*, Volume 10, Issue 4, Oct-Dec. 2003. Pages 60 – 70.
3. ContentGuard, Inc. "eXtensible rights Markup Language (XrML) 1.2 Specification", November 20, 2001. www.xrml.org.
4. ContentGuard, Inc. "eXtensible rights Markup Language (XrML) 2.0 Specification", November 20, 2001. www.xrml.org.
5. ContentGuard, Inc. "MPEG REL Resources". www.contentguard.com/MPEGREL_reference.asp.
6. ContentGuard, Inc. "XrML Use Cases", November 20, 2002. www.contentguard.com/Cookbook.asp.
7. Digital Media Project. www.dmpf.org.
8. DRM Watch. www.drmwatch.com.
9. IEEE Learning Technology Standards Committee. ltsc.ieee.org.
10. ISO/IEC 21000-5:2004. Information technology -- Multimedia framework -- Part 5: Rights Expression Language, 2004.
11. ISO/IEC 21000-6:2004. Information technology -- Multimedia framework -- Part 6: Rights Data Dictionary, 2004.
12. D. Kundur, C.-Y. Lin, B. Macq, and H. Yu (editors). Special Issue on Enabling Security Technologies for Digital Rights Management, *Proceedings of the IEEE*, Volume 92, Issue 6, June 2004.
13. C-C J. Kuo, T. Kalker, and W. Zhou (editors). Special issue on Digital Rights Management, *IEEE Signal Processing Magazine*, Volume 21, Issue 2, Mar 2004.
14. Microsoft, Windows Rights Management Services (RMS). "XrML", August 13, 2004. http://www.microsoft.com/resources/documentation/windowsserv/2003/all/rms/en-us/TechRef_46.msp.
15. M. Miron, T. DeMartini, X. Wang, and B. Gandee. "The Language for Digital Rights", presented at MPEG 58th Meeting, Pattaya, Thailand. December 2, 2001. www.xrml.org/reference/MPEG_Thailand.ppt.
16. The Moving Picture Expert Group (MPEG). www.chiariglione.org/mpeg/
17. OASIS eXtensible Access Control Markup Language (XACML) Technical Committee. <http://www.oasis-open.org/committees/xacml>.
18. OASIS UDDI Technical Committee. UDDIV3, *UDDI Version 3.0*. July 19, 2002. uddi.org/pubs/uddi-v3.00-published-20020719.htm.
19. OASIS Web Services Security (WSS) Technical Committee. www.oasis-open.org/committees/wss.
20. Open Digital Rights Language Initiative. www.odrl.net.
21. Open EBook Forum. www.openebook.org.
22. Open Mobile Alliance, www.openmobilealliance.org.
23. RealNetworks. Extensible Media Commerce Language, www.xmlcl.org.

24. B. Rosenblatt, B. Trippe, and S. Mooney. *Digital Rights Management: Business and Technology*. Wiley; 1st edition, November 2001.
25. G. Rust, and B. Bide. "<indec> Final Report", 2000. <http://www.indec.org/project.htm#finalDocs>.
26. P. Samarati and S. De Capitani di Vimercati. "Access Control: Policies, Models, and Mechanisms", in *Foundations of Security Analysis and Design*, R. Focardi and R. Gorrieri (eds), LNCS 2172, Springer-Verlag. sansone.crema.unimi.it/~samarati/Papers/sam-fosad.ps.
27. R. Shirey. "Internet Security Glossary", IETF RFC 2828, May 2000. www.ietf.org/rfc/rfc2828.txt.
28. M. Stefik. "Letting Loose the Light: Igniting Commerce in Electronic Publication", in *Internet Dreams: Archetypes, Myths, and Metaphors*. Cambridge, MA: MIT Press, 1996.
29. W3C, Extensible Markup Language (XML) 1.0. February 10, 1998. www.w3.org/TR/1998/REC-xml-19980210.
30. W3C, Namespaces in XML, 14 January 1999, www.w3.org/TR/1999/REC-xml-names-19990114.
31. W3C, *Web Services Description Language (WSDL) 1.1*. March 15 2001. www.w3.org/TR/2001/NOTE-wsdl-20010315.
32. W3C, XML Schema. May 2, 2001. www.w3.org/TR/2001/REC-xmlschema-1-20010502 and www.w3.org/TR/2001/REC-xmlschema-2-20010502.
33. X. Wang and T. DeMartini. "XrML's Role in Information Piracy Prevention and Privacy Protection", *Computer*, Volume 36, Issue 7, July 2003. Page 74.
34. X. Wang, T. DeMartini, B. Wragg, M. Paramasivam, and C. Barlas. "The MPEG-21 Rights Expression Language and Rights Data Dictionary", to appear on IEEE Transactions on Multimedia. Volume 7, Number 3, June 2005.
35. X. Wang, G. Lao, T. DeMartini, H. Reddy, M. Nguyen, and E. Valenzeula. "XrML – eXtensible rights Markup Language". *Proceedings of the 2002 ACM Workshop on XML Security*. Fairfax, VA, USA. Pages 71-79.
36. Xerox Corp. "Digital Property Rights Language, Manual and Tutorial - XML Edition", Version 2.00, November 13, 1998. xml.coverpages.org/DPRLmanual-XML2.html.
37. XML Cover Page. "XML and Digital Rights Management (DRM)", xml.coverpages.org/drm.html.