Translation of Rights Expressions

Brenton Cooper
Motorola Global Software Group Australia
2 Second Avenue, Mawson Lakes SA 5095
brenton.cooper@motorola.com

Paul Montague
Motorola Global Software Group Australia
2 Second Avenue, Mawson Lakes SA 5095
paul.montague@motorola.com

Abstract

Digital Rights Management (DRM) is concerned with the controlled distribution and usage of digital works in order to prevent unauthorized usage of digital content. A significant component of a DRM system is a well-defined language for expressing the rights which a user/device has been granted. Such a language is known as a Rights Expression Lanugage (REL).

Various technical communities have developed alternative rights expression languages. It is expected that in the marketplace different rights expression languages will be used as determined by the business needs and constraints of content and service providers. From such market fragmentation arises the need to move content between different DRM schemes, and so there is a need to translate rights expressions between RELs.

In this paper, we examine the challenges that must be addressed when developing a rights expression translation and then present several approaches to rights expression translation.

Keywords: Digital Rights Management, Rights Expression Language, Translation, import, export.

1 Introduction

Technical communities such as the Motion Picture Experts Group (MPEG) and Open Mobile Alliance (OMA) have identified Digital Rights Management as a key enabler for future multimedia devices and applications. To that end, they along with other organisations have developed rights expression languages in which it is possible to express the rights which can be assigned to users for digital content.

It is likely that various DRM schemes and hence various RELs will co-exist in the marketplace. For example, digital rights management for content purchased via mobile devices (i.e. mobile phones) may be enforced by the OMA DRM scheme [OMA-DRM-DRM], whilst content purchased over a cable TV system may be controlled by the MPEG-21 DRM scheme [MPEG-21].

The user may well consider it inconvenient to have "islands of content" tied to the distribution system on which the content was bought. Why can the user not seamlessly move content from one system to another? For example, if the user has purchased and downloaded a video clip on her phone, but is now sitting in front of her television, it should be possible to acces the very same content. To enable this, whilst still enforcing the rights it is at least necessary to translate rights expresisons from one REL to another.

In this paper we present an overview of common rights expression languages, and then discuss the stages in developing a translation process between RELs.

2 Rights Expression Languages

Three common rights expression languages are the Open Digital Resource Language (ODRL), eXtensible Rights Markup Language (XRML) and the Open Mobile Alliance (OMA) Rights Expression Language. In this section we present a brief introduction to each of these RELs.

2.1 ODRL

The ODRL specification is presented in two sections,

- ODRL Expression Language – which defines the formal model for the rights expression language, often using abstract concepts (rights, permission, constraint etc)
- ODRL Data Dictionary – which defines the semantics of the 'concrete' elements which are used to express an instance of a rights specification (e.g. play, pre-pay etc).

2.1.1 ODRL Expression Language

ODRL is based upon an extensible model for rights expression which involves a number of core entities and their relationships. A simplified view of the ODRL Foundation Model is presented in Figure 1.

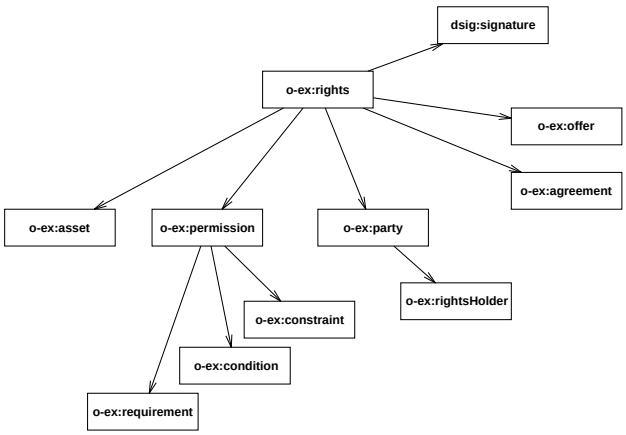

Figure 1 ODRL Foundation Model
(represented as a UML class diagram)

The three core entities in the model are

- *Assets* – which include any physical or digital content. The assets must be uniquely identified. The assets may consist of many subparts, be in many different formats and may be encrypted to enable secure distribution of content.
- *Rights* include Permissions which can then contain Constraints, Requirements and Conditions. *Permissions* are the actual usages or activities allowed over the assets (e.g. Play a video). *Requirements* are the obligations needed to exercise the permission (e.g. pay $5 each time you play the video). *Constraints* are limits placed upon the permissions (e.g. play a video a maximum of 5 times). *Conditions* specify exceptions that, if they become true, extinguish the permissions (e.g. if credit card expires then all permissions are withdrawn).
- *Parties* – include end users and rights holders. Parties can be humans, organisations and defined roles. End users are usually the asset consumers. Rights Holders are usually parties that have played some role in creation, production, distribution of the asset, and can assert some form of ownership over the asset and/or its permissions. Rights Holders may also receive royalty payments.

The ODRL Foundation Model uses the three core entities to express *Offers* and *Agreements*. Offers are proposals from Rights Holders for specific rights over their assets (e.g. distributor offers for sale an audio file for a flat payment of $5). Agreements are the transformation or acceptance of an offer into a licence for rights over assets by parties. Offers and Agreements are an important aspect of ODRL, as they clearly identify the purpose for which the rights expression has been generated.

In this section we have presented a high level overview of the central concepts within ODRL Expression Language. There are many more details which can only be appreciated by close inspection of the ODRL specification and schema [ODRL]. In particular, the ODRL Expression Language schema has been constructed such that for many types it allows recursive definitions e.g. a constraint can contain from one to many actual constraint elements, by utilising nested elements. Such recursion presents difficulties when contemplating translation to another rights expression language.

2.1.2 ODRL Data Dictionary

Many of the entities defined in the ODRL Expression Language are "abstract" – such as permissions, constraints, conditions, requirements. The ODRL Data Dictionary defines the concrete entities which may be used in the specification of rights. For example, while the ODRL expression language defines the o-ex:permissionElement element, the ODRL Data Dictionary defines actual elements such as o-dd:play, o-dd:print, o-dd:display, o-dd:lend, o-dd:sell, o-dd:backup etc. which can be substituted for the more generic permission. Substitution occurs via the use of XML schema element substitution groups.

2.2 XrML

The XrML 2.0 specification is split into several parts

- A core schema containing definitions of concepts that are at the heart of the semantics of XrML,
- A standard extension schema containing definitions of concepts which are generally and broadly applicable to XrML 2.0 usage scenarios, but not at the heart of the semantics, and
- A content specific extension schema which defines concepts which are specifically related to digital content or works.

2.2.1 XrML Core Schema

The most important concept within XrML is that of a *License*. A *License* is conceptually a container of *Grants*, each of which conveys to a particular *Principal* the sanction to exercise some identified *Right* against some identified *Resource*, possibly subject to the need for some *Condition* to be first fulfilled. The relationships between these core entities is shown in Figure 2.

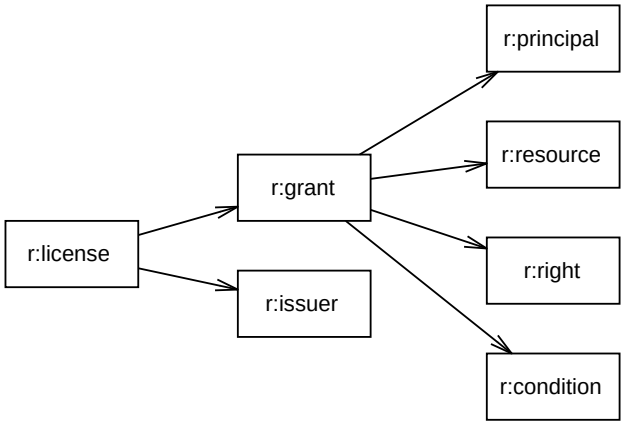

Figure 2 Simplified view of the XrML core schema (shown as a UML class diagram)

Many elements in the XrML core, including principal, resource, right and condition are conceptually abstract. As such, these literal XML elements must not appear in a grant. Elements which are *substitutable* for these types must be used instead. The XrML core defines some elements which are substitutable for these elements, however, many more substitutable elements are defined in the standard and content extensions to the XrML core.

2.2.2 XrML Standard Extensions

The standard extension schema contains the definition of a set of condition extensions, payment extensions, naming extensions and revocation extensions.

The condition extensions define a set of commonly applicable conditions which may be substituted for the abstract r:condition defined in the XrML core. These conditions include many which are themselves and extension of sx:StatefulCondition, a condition which allows state information to be retained. This concept is useful for tracking usage, fees, time intervals etc.

The payment extensions define various elements which can be used to construct flexible payment models. The payment elements are used by the sx:fee condition. The payment extensions include sx:cash, sx:paymentFlat, sx:paymentPerUse and more.

Together with the r:possessProperty right, the resource sx:Name and its extensions allow licences to straightforwardly express authorised association of names with principals. The standard extension defines sx:commonName, sx:dnsName , sx:emailName and sx:x509SubjectName extensions to the type Name.

2.2.3 XrML Content Extensions

XrML core and standard defines some general use rights, whereas the XrML content extensions define rights specific to distributing and using DigitalWorks. These rights are grouped as Render, Transport, Derivative Work, File Management and Configuration rights. These rights define the actions that a prinicipal may exercise such as cx:play, cx:print, cx:copy, cx:edit, cx:read, cx:backup, cx:install and more.

The content extension schema also defines the resources cx:DigitalWork and cx:Simple DigitalWorkMetdata which can be used together to describe a digital work by identifying how to locate the resource as well as information such as the title, creator, publisher etc.

Finally, the content extension schema defines extensions to the core r:condition such as cx:Renderer, cx:Source, cx:Helper, cx:Destination and cx:Watermark. These conditions may, for example, be used to identify the system (software or hardware) that can must be used to display content.

2.3 OMA Rights Expression Language

The Open Mobile Alliance (OMA) Rights Expression Language [OMA-DRM-REL] has been defined expressly for rights management in a mobile environment. To that end the OMA Rights Expression Language defines

1. A mobile profile (i.e. subset) of ODRL. The mobile profile omits the offer, agreement, rights-holder, condition and requirement models from ODRL.
2. A data dictionary defining additional permissions and constraints beyond those provided by ODRL.

Importantly, the OMA REL profile is a greatly simplified subset of the complete ODRL. This is a significant benefit when contemplating translation to another REL.

3 Translation between Rights Expression Languages

Irrespective of the particular language used a rights expression generally defines a set of rules governing the use of some asset or resource (e.g. a music clip). The rules usually identify

- *what* the asset is (e.g. a book entitled “Alice in Wonderland”),
- *how* the asset may be used (e.g. displayed, printed),
- *when* the asset may be accessed (e.g. for 7 days from first access), and
- *who* can access or use the asset (e.g. Mary Smith).

The definition of a rights expression translation is a two-stage process. The first stage is to develop a conceptual “equivalence” model between the primary entities of both the source and target rights expression language. At this level the model maps the *who*, *what*, *when* and *how* of a rights expression between the source and target language. The equivalence model provides a framework for the complete translation identified in the second stage.

The second stage is to extend the equivalence model by developing a detailed mapping between the XML schemas which describe the source and target rights expression language. The detailed mapping will identify how each element, its’ children and attributes map from the schema for the source language into elements and attributes in the schema for the target language.

4 ODRL-to-XrML Equivalence Model

In this section we present a "conceptual level" equivalence model between ODRL and XrML. Such a conceptual equivalence model is only a first-level approximation of how a rights expression in ODRL would be mapped into XrML. The analogy described by the conceptual equivalence model will form the basis of the complete mappings presented in [Motorola-2004].

The equivalence model presented herein is only "conceptual" as it does not consider the detailed representation of ODRL elements as defined in the ODRL expression language schema. The "conceptual model" does not consider

- relationships between various elements defined in the ODRL expression language (e.g. we claim that an ODRL permission is "equivalent" to an XrML rights, however an ODRL permission can contain zero or more requirements, but there is no such allowance for an XrML right.),
- attributes of the various ODRL elements and how they map to XrML, and
- structural concepts within ODRL such as containers, sequences, linking and inheritance.

The ODRL-XrML equivalence model is presented Table 1. The equivalence model considers how the most significant entities from the ODRL Expression Language can be mapped into equivalent concepts in the XrML core. Note that in this view of the equivalence model we do not consider how elements from the ODRL Data Dictionary should be mapped. (for complete details of the ODRL DD mapping refer to [Motorola-2004]).

ODRL	XrML	Notes
o-ex:rights	r:license	Rights contain multiple o-ex:offer/o-ex:agree/o-ex:revoke elements. License contains multiple r:grants/r:grantGroups.
o-ex:offer	r:grant/r:obtain or r:grant	If a "check-out" process is required, then grant g must have rights element r:obtain, and the resource contained in *g* must be present and must be either a grant or a grantGroup, *g'*. The use of the obtain right is conceptually an "offer" or "advertisement" by the issuer of the grant *g*, to "sell" the grant *g'*. If a "check-out" process is not required then grant g will have resource, right etc set as per the rest of the mapping.
o-ex:agreement	r:grant	
o-ex:revoke	r:grant/r:revoke	The ODRL revoke element can be used to revoke offers, agreements, permissions etc. The XrML core and extensions only define a mechanism to revoke signatures. Insufficient semantics in XrML equivalent.
o-ex:asset	r:resource	
o-ex:permission	r:right	The structure of o-ex:permission and r:right elements do not match. The child elements of o-ex:permissionType include o-ex:permissionElement, o-ex:requirement, o-ex:condition, o-ex:constraint, o-ex:asset and o-ex:prmissionElement. This supports complex relationships between permissions, requirements, conditions, constraints and assets. A corresponding structure is not directly supported by XrML core. The o-ex:permissionElement is substituted by elements defined in the ODRL DD (e.g. o-dd:play). Those elements substitutable for o-ex:permissionElement are analogous to types defined in XrML content and standard extensions that derive from r:right (e.g. sx:play). Refer to [Motorola-2004] for mapping of the ODRL Data Dictionary.
o-ex:constraint	r:condition	The structure of o-ex:constraint and r:condition elements do not match. The child elements of o-constraintType include o-ex:constraint and o-ex:constraintElement. This supports nesting of constraints. A corresponding structure is not directly supported by the XrML core. The o-ex:constraintElement is substituted by elements defined in the ODRL DD (e.g. o-dd:count). Many of those elements substitutable for o-ex:constraintElement are analogous to types defined in XrML standard and content extensions that derive from r:condition (e.g. sx:Exercise:Limit). Refer to [Motorola-2004] for mapping of the ODRL Data Dictionary.
o-ex:requirement	r:condition	
o-ex:condition	N/A	XrML does not support this concept.
o-ex:revoke	r:grant/r:revoke	XrML revocation mechanism is more limited than ODRL. XrML allows the revocation of signatures only. ODRL allows the revocation of offers, agreements, permissions etc.
o-ex:party	r:principal	
o-ex:rightsholder	N/A	The ODRL concept of a rightsholder is not directly transferable to XrML. However, it is possible to specify XrML conditions which capture the royalty payments in some situations.
o-ex:context	N/A	The context element contains information which is interpreted based upon the type of its parent element. There is no immediate correspondence with a single XrML element, rather the sub-elements are considered only in the detailed mapping (refer to [Motorola-2004] for more details).

Table 1 ODRL-to-XrML Equivalence Model

5 Challenges for Translation

Whilst the development of a conceptual equivalence model is relatively straightforward, a complete translation between XML schema for the source and target language is more complicated. In this section we detail some of the challenges faced when developing a translation model from ODRL to XrML.

5.1 Unsupported Concepts

Unsupported concepts are situations in which a concept defined within ODRL has no analog in XrML. This includes,

- *o-ex:rightsholder* - ODRL states that a "rights holder is a recognised party and any set of entitlements that are due for their asset". The ODRL examples provided in the specification [ODRL] predominately use the rights holder element to express royalty payments, however the inclusion of royalty terms is optional. The ODRL specification is unclear as to the intended interpretation of a rights holder element which has no royalty information. When mapping to XrML, it may be possible to translate the "royalty" component of a rightsholder element into equivalent XrML conditions (e.g. sx:fee), however, there is no equivalent way of expressing that a designated principal has some form of ownership as is implied by the ODRL rights holder.
- *Nesting* - ODRL supports nesting of the same type. For example, an o-ex:constraint can have multiple child elements of type o-ex:constraint. This concept is not supported within the XrML core, standard or content extensions. ODRL also allows a very complex nesting of different types e.g. conditions can contain permissions which contain requirements which contain constraints. In general it is not possible to map this structure directly onto the equivalent entities in XrML.
- ODRL supports *expression containers*, which are a mechanism for grouping collections of permissions, constraints, conditions or requirements. Each container is identified as being of type 'and', 'or' or 'exclusive or'. Whilst XrML does support some elements which display container like properties, such as the r:allConditions and r:allPrincipals elements, there is no mechanism for translating a generalised container into XrML.
- ODRL also supports *expression sequences* which is a mechanism for specifying the order in which the entities must be addresses. The allowed orderings are 'partial' and 'total'. There is not an equivalent concept within XrML.
- ODRL supports the ability to specify the *inheritance* of expressions between assets. There is no mechanism to do this in XrML.

5.2 Implementation Differences

In this section we present situations in which there exists an XrML equivalent to the original ODRL concept, however, the mapping between the two concepts is difficult due to significant implementation differences.

- *Stateful conditions* - ODRL specifies only the initial or maximum value of a stateful condition (e.g. the o-dd:range and o-dd:count constraint elements). ODRL does not specify how and where the state should be maintained and queried. In contrast XrML specifies only the location of a "service" which maintains the state. The service is to queried to determine if the rights should be granted. These two mechanisms for implementing stateful conditions are not directly equivalent.
- *Multiplicity* - ODRL type definitions routinely allow between zero and infinite instances of each of the child elements. For example, an o-ex:offer element may contain from 0 to infinite child elements of type o-ex:permission. In contrast, an XrML r:grant element may optionally contain only a single r:right element. Such differences in representation require 1-to-many translations.

  It is worthwhile noting two special elements within XrML, namely the r:AllPrincipals and the r:AllConditions types. These types are containers of zero or more Principals and Conditions respectively. Whilst a single grant can have at most one child r:condition element, that r:condition element may be an r:allConditions element, which itself can contain many r:conditions. Similarly, whilst a grant can contain at most one child r:principal element, it may be an r:allPrincipals element. These two concepts allow a single grant to effectively have multiple conditions and principals just as is possible in the equivalent ODRL offer or agreement.
- *o-ex:revoke* - ODRL allows the revocation of offers, agreements, permissions etc, whilst XrML allows only signatures to be revoked. Consequently, there is no generalised way in which to directly translate the o-ex:revoke type into XrML.
- *o-ex:context* - ODRL does not provide definitive rules for use of the context element and its children. The context element is appears as a child element for almost all types in the ODRL expression language. It is possible that only a specific application will define how the context element and its children should be used, and what the meaning for each sub-element should be. Consequently, in most places in which the context element is used it is possible to translate only a select number of its children to equivalent elements in XrML.

5.3 Semantic Mismatch

In this section we present situations in which there exist subtle semantic differences between otherwise seemingly equivalent ODRL and XrML concepts.

- *Multiplicity* - It should be noted that XrML places a more strict semantic interpretation on multiple principals than ODRL does for multiple parties.

An XrML r:allPrincipals element semantically represents the logical conjunction of the principals represented by all its children. That is the r:allPrincipals element represents all the principals acting together as a single entity. For example, as stated in the XrML specification [XrML] "if an allPrincipal *a* is identified in a grant as the principal which must sign a certain bank loan application, then, conceptually it is being required that each of the children of *a* act together as co-signers of the loan application.

ODRL does not place a semantic meaning on the use of multiple parties in a rights expression. By mapping multiple parties into a r:allPrincipal element, then rightly or wrongly we will have introduced additional meaning to the rights expression.

5.4 Ambiguities

Ambiguities are situations in which the ODRL definition is sufficiently vague so as to make it impossible to develop a suitable translation for the ODRL concept.

- *Sequence and container items* - ODRL appears to have some limitations in its syntactical definition. e.g. the elements o-ex:container and o-ex:sequence routinely appear as children of other elements. The container and sequence types are generalised collections that imply a certain ordering over the items in the collection, however, almost all types specified within the ODRL Expression language are permitted to be present in the collection. This creates difficulties when attempting to understand the intended meaning/usage of certain types defined within ODRL. For example, the type o-ex:requirementType has a child element of type o-ex:sequenceType and o-ex:containerType. Now, in this context the items in these collections would presumably be requirements or requirementElements, however they are also permitted to be conditions, constraints, permissions and rightsHolders. I suspect that this is not intended by the authors of the ODRL, however, it does seem to be a legal use of the XML schema.

6 Techniques for Translation

In this section we present two alternative techniques for translation of rights expression languages.

6.1 Direct Translation

For each element in the source language schema, the direct translation technique establishes a mapping to a corresponding entity in the target language. It should be noted that,

1. The translation must identify how each attribute and child element is mapped into corresponding entities in the target language schema.
2. The mapping need not be one-to-one, indeed it can be many-to-many.
3. Not all elements or attributes in the source language have a corresponding entity in the target language. For those entities which do not map to the target language, they are omitted in the translation process. Consequently the translated rights expression language will not contain all information present in the original rights expression language. That is to say that the translation process is *lossy*. Note that if the original rights expression could be limited to contain only those entities which map to the target language then the translation process would not cause loss of information. Such a limited input syntax could be represented as a *profile*.

A high-level view of the direct translation mechanism is shown in "Figure 3. UML package diagram depicting relationships between XML schemas for direct translation mechanism" From this diagram it can be seen that the direct mapping

- translates elements defined in the ODRL Expression Language (ODRL–EX) to elements defined in the XrML core, and
- translates elements defined in the ODRL Data Dictionary (ODRL-DD) to elements defined in the XrML core, XrML standard extensions and XrML content extensions.

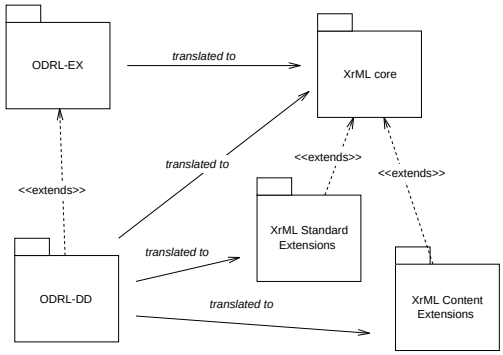

Figure 3. UML package diagram depicting relationships between XML schemas for direct translation mechanism.

A direct translation of ODRL to XrML is presented in [Motorola-2004]. The direct translation is not presented here for sake of brevity.

6.1.1 Profiles

During the direct translation process it is often necessary to omit elements of the source right expression as there is no conceptually equivalent element in the target language. Such omitted information results in a "lossy" translation. In many instances this would be unacceptable. One possible solution is to limit the elements contained in the source language rights expression such that only those elements which have a conceptual equivalent in the target language may be used. The definition of a limited subset of the source rights expression language is called a "profile".

The Open Mobile Alliance (OMA) Rights Expression Language [OMA-REL] defines a profile of ODRL for use in the mobile environment, as well as some extensions to the basic ODRL. Importantly, the OMA REL profile does

not include many aspects of ODRL which may be considered as "troublesome" when attempting to map to XrML. For instance, the OMA REL profile

- does not include containers and sequences,
- greatly simplifies valid rights expressions by restricting the multiplicity of many elements,
- severely constrains the permissible data dictionary elements.
- eliminates the requirement, condition and revocation models, and
- provides a well-defined yet limited context model.

Consequently, it is much easier to define a direct translation for the OMA REL profile than it is for the entire ODRL. A direct translation for the OMA REL profile is also presented in [Motorola, 2004]. Again, size constraints prevent the complete translation being presented here.

The advantages of adopting a "profile" for direct translation are that

- no information from the source language rights expression is lost in the translation process, as all elements have an analgous representation in the target language, and
- the resultant rights expression uses only elements from within the target language, so devices which can interpret the target language do not need to be modified to understand a translated rights expression.

6.2 Intermediate Schema

The direct translation method proposed in the previous section often needs to omit information from the source language rights expression during the translation process. One of the primary reasons for this is that the structure of the source and target rights expression languages do not match. This was highlighted in Section 5.1, where the inability for XrML to match the very general nesting possible within ODRL rights expressions was discussed.

To overcome this inability, it is possible to introduce an intermediate schema which uses the central elements frm the target language (in this case XrML), but within this new schema the element-to-element relationships are such that they mimic the structure of the source language. For example, in ODRL it is possible for a o-ex:rights element to contain multiple o-ex:permissions and multiple o-ex:requirement elements. To mimic this structure in the intermediate schema we would ensure that a r:grant element may contain multiple r:rights and multiple r:conditions. In this way the intermediate schema can mimic the structure of the source language, but use "concepts" from the target language.

When compared to the direct translation technique, the major advantage of developing an intermediate schema is that more information from the source language rights expression can be translated into the intermediate schema. However, to interpret the intermediate schema requires modifications to the target devices.

7 Conclusions

The need to translate rights expressions between different rights expression languages arises from the user's need to move content between platforms.

Unfortunately, when adopting a straightforward "direct translation" model the act of translating rights expressions often causes information to be lost during the translation process. It seems that the most sensible approach to rights translation is to use a profile, i.e. to limit the elements contained within the source language rights expression to only those that have a conceptual equivalent in the target language.

8 Acknowledgements

This work has been supported by the Smart Internet CRC, in particular Rei Safavi-Naini, Nicholas Sheppard and Takeyuki Uehara.

9 References

Motorola Global Software Group Australia, *Translation of Rights Expression Langauges: ODRL to XrML*, 2004.

ODRL, *Open Digital Rights Language* (2002), Version 1.1, 2002-08-08, Renato Iannella (editor).

Open Mobile Alliance (OMA), *DRM Rights Expression Language*, Candidate Version 2.0 – 16 July 2004, OMA-DRM-REL-V2_0-20040716-C.

Open Mobile Alliance (OMA), *DRM Specification*, Candidate Version 2.0 – 16 July 2004, OMA-DRM-DRM-V2_0-20040716-C

ContentGuard, eXtensible Rights Markup Language (XRML) 2.0 Specification, 20 November 2001.

MPEG21-Part1, *Multimedia Framework – Part1: Vision, Technologies, and Strategy*, ISO-IEC TR 21000-1:2002.