Preface:

Welcome to the Second International Open Digital Rights Language (ODRL) Workshop. ODRL is an XML-based rights expression language (REL). A rights expression language is a means of expressing usage and access rights of parties to assets. Rights expression languages provide a syntax and semantics that are sufficiently rich to formulate rights expressions for any kind of digital media content, such as digital publications, audio and video files, images, games, software, and other digital or physical goods, including pricing models as well as terms and conditions, regardless of whether a monetary consideration is part of the transaction. Consequently, rights expression languages provide a metadata framework for the expression of rights.

The ODRL Initiative has gained international significance in the field of digital rights management (DRM) over the past years, culminating in ODRL being adopted as an international standard by the Open Mobile Alliance for supporting the process of mobile content distribution and management. The objective of the ODRL International Workshop is to bring together the research and industry communities to share experiences and discuss the future developments of the ODRL language and to ensure its timeliness, usability, openness, and future success.

ODRL is seen as key infrastructure element for the management and trading of content in the digital environment. ODRL enables the formulation of machine readable, interoperable contracts between rights holders and content users and need to evolve as the community awareness increases and business models change. The role of the Workshop is to enable this process.

The workshop would not have been the success we trust it will be without the support of the two hosts ADETTI and ISCTE and our sponsors, LiveEvents Wireless, NICTA and o2 Germany. We also would like to thank the members of the Program Committee for their work refereeing the papers. A special mention goes also to the Organizing Committee for their constant efforts in making possible that this event could take place.

- ODRL Iniciative
- ADETTI - Associação para o Desenvolvimento das Telecomunicações e Técnicas de Informática
- ISCTE - Instituto Superior de Ciências do Trabalho e da Empresa
- LiveEvents Wireless
- NICTA

Lisbon, July 2005

Carlos Serrão
Renato Iannella
Susanne Guth

Workshop Chairs and Editors
http://odrl.net/workshop2005/

July 2005
ISBN: 1-740-64501-4

Editorial Boards:

Editor in Chief

Carlos Serrão (ADETTI / ISCTE)

Editors

Renato Iannella (National ICT Australia [NICTA]), Susanne Guth (O_2 [Germany] GmbH), and Carlos Serrão (ADETTI / ISCTE)

Editorial Production

Carlos Serrão (ADETTI / ISCTE), Ana Rita Leitão (ADETTI) and Frederico Figueiredo (INESC) .

Organisation Committee

Carlos Serrão (ADETTI / ISCTE), Miguel Salles Dias (ADETTI / ISCTE), and Ana Rita Leitão (ADETTI).

Program Committee

- Jaime Delgado - Universitat Pompeu Fabra, Spain
- Stephane van Hardeveld - VirtuosoMedia, The Netherlands
- Eckhart Koeppen - Nokia, Finland
- Magda Mourad - IBM T.J. Watson Research Center, USA
- Xavier Orri - Octalis S.A., Belgium
- Olli Pitkanen - Helsinki Institute for Information Technology, Finland
- Steve Probets - Loughborough University, UK
- Mark Strembeck - Vienna University of Economics and Business Administration, Austria

July 2005
ISBN: 1-740-64501-4

2nd International ODRL Workshop 2005

Lisbon, Portugal — 7-8 July 2005

Workshop Hosts

Workshop Program - FINAL

Instituto Superior de Ciências do Trabalho e da Empresa

Workshop Sponsors

Location: Auditorio 5 [ISCTE - Ala Autonoma - Building 2]

Thursday 7 July 2005

09:00-09:20	Workshop Registration
09:20-09:30	Workshop Welcome and Opening *Renato Iannella, Workshop General Chair* *Susanne Guth & Carlos Serrao, Workshop Program Chairs*
09:30-10:30 Keynote	Identity and Content Rights *Simon Nicholson, Director, Wireless Business Strategy & Development, Sun Microsystems, USA*
10:30-11:00	Break
11:10-11:30 Paper	A Review of the OMA DRM V2 ODRL Profile *Renato Iannella, NICTA, Australia*
11:30-12:10 Paper	A new Approach for Interoperability between ODRL and MPEG-21 REL *Jaime Delgado, Jose Prados, and Eva Rodriguez, Universitat Pompeu Fabra, Spain*
12:10-12:40 Paper	Embedding ODRL Statements in Dublin Core *Enric Peig and Jaime Delgado, Universitat Pompeu Fabra, Spain*
12:40-14:00	Lunch
14:00-14:40 Paper	Using ODRL to express rights for different content usage scenarios *Carlos Serrao, Miguel Dias and Jaime Delgado, Adetti/ISCTE, Portugal and Universitat Pompeu Fabra, Spain*
14:40-15:20 Paper	Formalising ODRL Semantics using Web Ontologies *Roberto Garcia, Rosa Gil, Isabel Gallego and Jaime Delgado, Universitat Pompeu Fabra, Spain*
15:20-15:50	Break
15:50-16:30 Open Panel	The impact of DRM Patents on REL Research and Standards *Susanne Guth, O2, Germany* *Renato Iannella, NICTA, Australia*
16:30-17:30 Keynote	DRM Coordination work on IST FP6 NAVSHP projects *Prof Miguel Dias, Adetti President, ISCTE Lecturer, IST-FP6 MEDIANET Project, Portugal*
20:00-22:30	Workshop Dinner Adega do Kais Restaurant Cais da Viscondessa, Rua da Cintura - Santos, 1200 - 109 LISBON

Friday 8 July 2005

09:00-10:00 Keynote	OMA DRM 2.0 Status and Future Work *Jan van der Meer, OMA DRM WG Leader & Philips Electronics, Netherlands*
10:00-10:30 Invited Talk	Plans, Scope, and Objectives of the GeoDRM WG within the Open Geospatial Consortium (OGC) *Roland Wagner, Universitat Munster, Germany*
10:30-11:00	Break
11:00-11:40 Paper	Extending ODRL to Enable Bi-Directional Communication *Alapan Arnab and Andrew Hutchison, University of Cape Town, South Africa*
11:40-12:20 Paper	Predicting the evolution of digital rights, digital objects and DRM languages *Jonathan Schull, Rochester Institute of Technology, USA*
12:20-12:50 Open Discussion	ODRL Initiative future directions and Working Group plans *Workshop Chairs*
12:50-13:00	Workshop Wrapup and Closing

A Review of the OMA DRM V2 ODRL Profile

Renato Iannella, *National ICT Australia (NICTA), Australia <renato@nicta.com.au>*

Abstract: This paper presents a review of the OMA DRM Version 2.0 profile of the ODRL REL. It looks at the decisions made by the OMA DRM working group and offers alternatives. The lessons are important to both the ODRL Initiative and to other groups developing profiles of the ODRL REL.

Index terms - ODRL, DRM Open Mobile Alliance, REL Profile

I. INTRODUCTION

The OMA DRM Version 2.0 specification [3] extended the profile adopted in the OMA DRM Version 1.0 specification [2]. The extensions included new elements - specific to the OMA community - and reuse of some of the standard ODRL data dictionary elements. A summary is shown in the Table below.

Permissions	Constraints
Play	Count
Display	Timed-Count *(OMA extension)*
Execute	Datetime
Print	Interval
Export *(OMA extension)*	Accumulated
	Individual
	System *(OMA Extension)*

This paper will review some of these elements and analyse the different options and issues in creating XML profiles of ODRL. See [1] for a discussion of the use of XML in ODRL.

II. THE COUNT ELEMENT

The OMA DRM REL required a more refined version of the "count" constraint. The requirement was to allow for a period of time to elapse before decrementing the counter. The rationale being that, in come cases such as audio media, the act of playing is not registered until a "few seconds" into the track. This would allow consumers to stop playing the track without effecting their count constraint if they are within this small period of time.

OMA decided to create a new Constraint to capture this refined count constraint. Alternatively, they could have extended the existing Count constraint, and hence, kept some level of interoperability (and backward compatibility) with other ODRL implementations, including OMA DRM version 1.0 systems.

All constraints can have any attribute from any other XML namespace (as defined in the XML Schema). Hence, OMA DRM could have just defined an additional attribute in the schema profile, such as:

```
<xsd:attribute name="timer"
      type="xsd:positiveInteger"/>
```

and then used this with the standard ODRL count element, such as:

```
<o-dd:count oma-dd:timer="30"> 10
      </o-dd:count>
```

Another option could have been to utilise the standard "type" attribute that can appear on all constraint elements. You would then need to de-

fine the structure of the URI value for “type” such as a URN prefix (eg "oma:reduce-state:") followed by a positive integer of seconds. For example:

```
<o-dd:count o-ex:type="oma:reduce-state:30">
    10 </o-dd:count>
```

The introduction of the timer element could also cause some confusion with different permissions. The OMA DRM WG recognised this with the export and print permission and explicitly disallow its use. However, its use with Display is unclear, as display will render static content only. The timed-count permission should have been limited to only time-based media content (eg audio, video, games).

III. THE EXPORT ELEMENT

The Export permission allows users to convert the content to other formats, and is aimed at supporting future interoperability and maximising the applicability of content across platforms. There are two attribute “modes” defined that control the export:

- move - the content is moved from the original device to another device and deleted from the original
- copy - the content is copied from the original device to another device and kept on the original device

The semantics of the export are very similar to that of ODRL’s “move” and “duplicate” permissions. For example, the following two elements would be equivalent:

```
<oma:export mode="move">
<o-dd:move>
```

and:

```
<oma:export mode="copy">
<o-dd:duplicate>
```

The issue here would be the trade-off between the level of equivalence of these statements, and wider interoperability.

IV. THE SYSTEM ELEMENT

The System element is a Constraint that is used by the Export permission. It constrains the export operation to specific “systems”. This maybe useful in ensuring that exported content only moves to platforms that can support DRM, although that is not mandatory.

There are a number of existing ODRL elements that could be used here:

- cpu - any system with a cpu
- hardware - any generic hardware device
- software - any software dependencies

The question would be what types of "exports" are envisaged with OMA DRM 2.0 and how can they be controlled?

Likely candidates include:

- other physical mobile devices
- other physical desktop devices
- specific (DRM) platforms must be present

Also, this process may include "conversion" of the content and Rights Object to another platform (eg Real Helix, Microsoft Media).

So there maybe a need to control both aspects of the "export" - the type of device and the platform - together and individually.

For example, to limit to other hardware, the following may suffice:

```
<o-dd:duplicate>
  <o-ex:constraint>
    <o-dd:hardware>
      <o-ex:context>
        <o-dd:uid>oma:apple:ipod</o-dd:uid>
      </o-ex:context>
    </o-dd:hardware>
  <o-ex:constraint>
</o-dd:duplicate>
```

And to limit to platforms:

```
<o-dd:move>
  <o-ex:constraint>
    <o-dd:software>
      <o-ex:context>
        <o-dd:version> 7.0 </o-dd:version>
        <o-dd:uid> oma:real:helix </o-dd:uid>
      </o-ex:context>
    </o-dd:software>
  <o-ex:constraint>
```

</o-dd:move>

The issue of exporting requires greater analysis as to the many options that content maybe exported to. Then the semantics can be further refined by reusing ODRL terms and potentially defining new semantics.

V. INHERITANCE MODEL

The inheritance model adopted by OMA DRM is aimed at supporting the "subscription" business model. It uses a Parent Rights Object (RO) as the key to any subsequent Child ROs that are delivered to the device. In effect, you need the Parent RO for the Child RO to "inherit" from - and this requirement - is mapped into being a member of a subscription service.

This is an interesting take on the original need for an inheritance model in ODRL. This was based on the more traditional need to generically inherit rights from other rights statements.

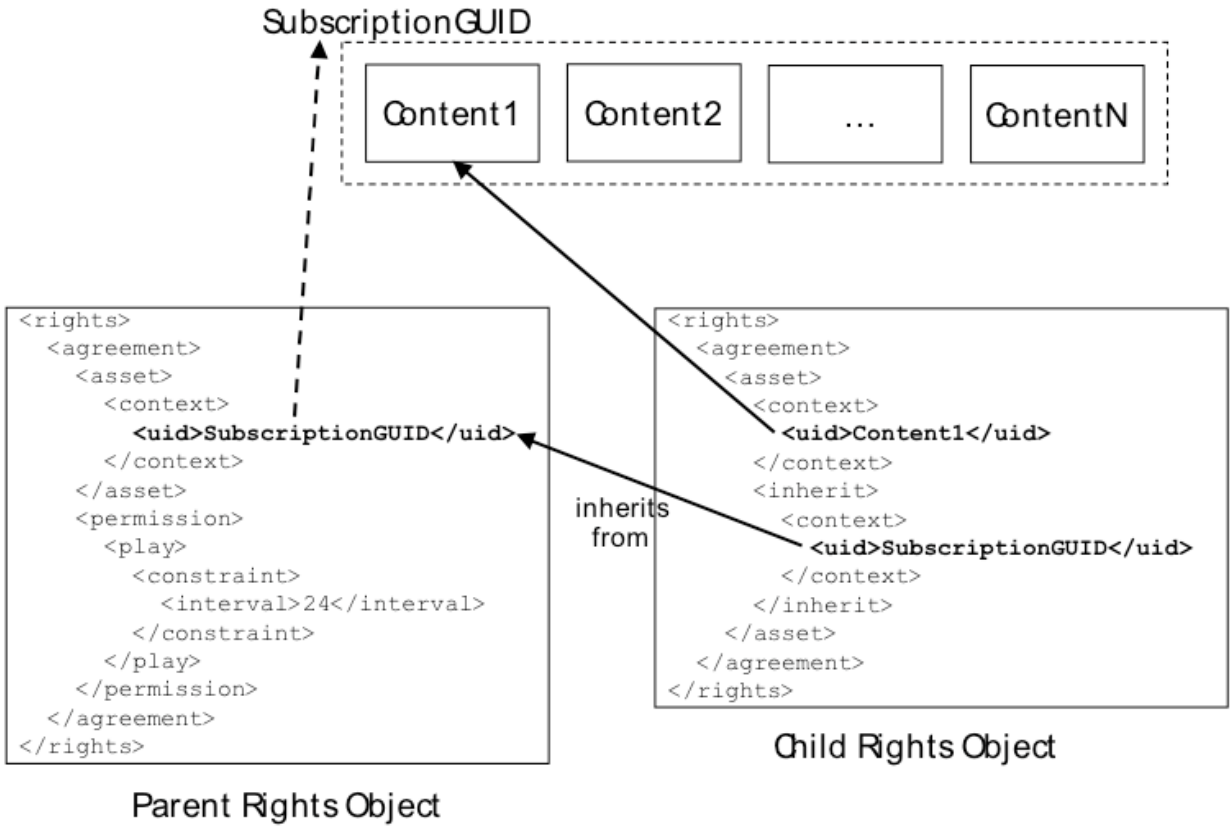
The OMA DRM view on the inheritance model is to overload the UID element. In this case, if the UID refers to some "virtual" content, and the Child ROs inherit from the same UID, then we have a subscription model. See the Figure below taken from the OMA DRM REL Specification [4].

The SubscriptionGUID has special meaning and a DRM Agent must be able to detect this. This is helped by the fact that a Parent RO will not have a KeyInfo or Digest element, as there is no real content. However, the subscriptionGUID does "point" to a real thing - so one could argue that all the Content (from 1 to N) inside this thing - are all available from just the Parent RO only.

A number of questions arise from this subscription inheritance model:

- Where should the actual permissions be located?
- What about the current state of the inherited permissions?
- Is it needed at all?

There are three models as to where the actual permissions may be located. In the Parent RO, Child RO, or both. In most cases, the Child RO should contain the actual permissions, as this is "closer" to the content, and Parent RO is really being used as a further "check" that the client has previously subscribed to this service. This a little bit of "over kill" as a Child RO, as a normal RO, that is sent to a client, would require some back-end service knowing who has already subscribed and will be pushing out the subsequent content -



all cryptographically bound to the end device. Nevertheless, the use of inheritance does make the model seem more “realistic”.

Technically, the permissions may appear in any of the three model options above. Experience over time will tell if any of the models have greater benefits over the others.

Another issue deals with the current state of the rights expression. When you inherit, do you not only inherit the permissions, but their current state? Is this desirable as well? The ODRL specification is silent on this issue, but the original intent was that you only inherit the permissions, not their current state.

The example in Table 1 shows the Parent RO with an Interval constraint (we assume the value should be “P24H”). This may mean that the total subscription covers a 24 hour period - or that each part of the subscription covers a 24 hour period. What would happen if the “state” of the Parent RO was expired (ie a 24 hour period had passed since it was first used) and then a Child RO is received with no new permissions? It could be interpreted to mean that the whole subscription is finished (based on the Parent RO) or that you now have another 24 hours to play the new content.

The various options here will need to be more fully discussed. There probably are cases where the example described may be the desired outcome. (For example, the OMA DRM specification makes a clear decision that state is not copied when exporting ROs. The same maybe needed for inheritance.)

The last big issue is if inheritance is needed at all to support subscription. You could certainly use inheritance for the original idea of having common rights that other expressions can use to inherit from. But there is also nothing stopping a service from providing a subscription to content and simply sending ROs when appropriate to the client.

VI. UIDs AND VERSIONS

The specification uses both the UID and Version context elements to define which systems to limit any exports to. The Open Mobile Naming Authority (OMNA) - part of OMA - will publish formal identifiers for the various systems. However, there maybe some need to standarise on the version numbering as well. For example, even the simple difference between “10” and “10.0” may make a difference to the parsing of the version number. Even worse may be non-numerical version identifiers.

VII. OVERRIDING SEMANTICS

The specification indicates: “If the <export> permission is granted to more than one target system, then these are enumerated by using multiple <uid> elements. In this case, the <count> constraint applies to the combined export transactions of all target systems.”

This has overridden the normal semantics of ODRL. In the normal case, a count constraint would be “and-ed” with all the other constraints. So, a count of “1” for two “systems” would allow both to occur.

For the OMA DRM view to be expressed (in this particular case), you can use the Container construct with the “or” boolean between two system constraints.

We assume that by doing this (supporting the container model) would increase the complexity of processing the ROs.

VIII. GRACE PERIOD

The Interval and Accumulated constraints both must “stop the execution of the permission as soon as possible after the value of the element has elapsed” and that this “should happen immediately”. It is not clear why “as soon as possible” was included nor why the “should” is not a “must”.

There now seems to be some possible "delay" to the Accumulated and Interval constraints. And in some cases, this could be a “user friendly” issue and by offering some “grace period” would improve the DRM experience.

This could then be generalised with the "timer" attribute in <timed-count>. That is, have one attribute "oma:delta" that indicates the number of

seconds you can wait before the permission must stop or be recorded.

IX. PRIVACY ISSUES

The OMA DRM specification allows for content to be shared - forwarded from user to user via super-distribution. This transaction can be tracked by the implementation. The OMA DRM specification does not include tracking as part of the REL - even though that facility is available in the complete ODRL REL. In such cases "tracking" can become one of the ODRL Requirements, hence explicitly making this feature something that the consumer has to agree to before acquiring the content.

At the same time, this also make it clear to the consumer what will happen when then do acquire this content. Since OMA DRM leaves tracking to be an "implementation issue", it does not guarantee that consumers will be aware of this requirement. Worse, it creates an nebulous situation in which a consumer's actions can be reported without their clear knowledge. In some cases, this can lead undesirable outcomes for the end consumer.

OMA DRM should have included the "tracked" requirement in their REL profile. This would make it always clear to the consumer what they can expect, and ensure that their privacy is not compromised.

X. OTHER ELEMENTS

The new profile did not consider including the rights holder and payment information. The rights holders would have been useful to assert the true owners of the content and may then allow end users to be aware that there is such important information available.

Perhaps the most disappointing is the non-inclusion of payment information. To meet the long term goals of interoperable content services, there needs to be support for information on how payments are handled. This would enable content owners to provide packaged content+rights to many different service providers, and not have to deal with each individually on the terms and conditions for payments.

XI. CONCLUSION

This paper has reviewed some of the decisions made by the OMA DRM Working Group in developing the ODRL profile for version 2.0. It has provided some feedback towards different options that may have been available, as well as discussed some of the advantages and disadvantages of these decisions and raised some of the semantic issues.

Overall, I think this is an excellent use-case for all parties (OMA and the ODRL Initiative) and will help in future work and more specifically, future ODRL profiles for OMA DRM.

ACKNOWLEDGMENT

National ICT Australia (NICTA) is funded by the Australian Government's Department of Communications, Information Technology, and the Arts and the Australian Research Council through Backing Australia's Ability and the ICT Research Centre of Excellence programs.

REFERENCES

[1] Iannella, R. The Open Digital Rights Language: XML for Digital Rights Management . Information Security Technical Report, Volume 9, Issue 3, July-September 2004, Pages 47-55.

[2] OMA (2003). Open Mobile Alliance DRM Specifications, Version 1.0 Approved Enabler, November 2003 <http://www.openmobilealliance.org/release_program/drm_v10.html >

[3] OMA (2004). Open Mobile Alliance DRM Specifications, Version 2.0 Candidate Enabler, July 2004 <http://www.openmobilealliance.org/release_program/drm_v20.html >

[4] OMA (2004). Open Mobile Alliance DRM Rights Expressions Language Version 2.0, 10 Dec 2004 <http://www.openmobilealliance.org/release_program/docs/DRM/V2_0-20041207-C/OMA-DRM-REL-V2_0-20041210-C.pdf>

A new approach to interoperability between ODRL and MPEG-21 REL

Jaime Delgado, Jose Prados, Eva Rodríguez
Universitat Pompeu Fabra, Passeig de Circumval·lació, 8, 08003 Barcelona, Spain
{jaime.delgado, josep.prados, eva.rodriguez}@upf.edu

Abstract

A key issue for the real deployment of Digital Rights Management (DRM) systems is interoperability. A clear example is at the level of Rights Expression Languages (RELs), where two of them are a prominent role. On the one hand, ODRL (Open Digital Rights Language) is an initiative being used, for example, by the Open Mobile Alliance (OMA), a relevant industrial forum in the area of mobile and on the other hand systems MPEG-21 REL is an ISO/IEC standard. MPEG-21 REL is more complete, but rather complex although not exhaustive; this is why ODRL could be considered as a more flexible option.

In this paper, we analyse two DRM specifications from OMA, and try to propose its implementation in an MPEG-21 environment. In addition tools able to work in both environments are presented. By defining an MPEG-21 REL DTD, a minor extension of the MPEG-21 REL, and the use of the MPEG-21 IPMP (Intellectual Property Management and Protection), we are in fact specifying MPEG-21 REL profiles. This approach could simplify the implementation of MPEG-21 REL applications and facilitate its interoperability with ODRL. In order to verify the feasibility of our proposal, we have implemented some tools that work with both MPEG-21 REL and OMA DRM.

1. Introduction

In this paper we focus on the interoperability between Rights Expressions Languages, a clear key issue in order to achieve interoperability among complete DRM systems.

In [1] we presented a first approach to achieve interoperability between ODRL [2] and MPEG-21 REL [3]. In this first study we concluded that a syntactic approach to map licenses expressed in the two different languages would only be feasible for a subset of both languages, that could be identified as profiles.

As OMA (Open Mobile Alliance) [4] has developed the OMA DRM Rights Expression Language versions 1.0 [5] and 2.0 [6] based on ODRL, we have decided to define a specific subsets for MPEG-21 REL equivalent to those specified by OMA.

Therefore, in this paper we present how to achieve interoperability between MPEG-21 REL and ODRL for these specific subsets. The MPEG-21 REL subsets defined provides the same features as both OMA DRM RELs. For the first version presented by OMA, it is enough to restrict MPEG-21 REL to achieve interoperability, but for the second version of OMA DRM REL we also have to extend MPEG-21 REL as it does not provide all needed functionalities. OMA DRM REL v2.0 introduces the security and inheritance models that have not been considered in MPEG-21 REL. Then, we have extended the MPEG-21 REL to provide such functionalities. Nevertheless, in the case of security information, we have considered two approaches. In the first one, we have extended MPEG-21 REL defining the appropriate elements to describe the tools that protect the content, while in the second approach, we have used the MPEG-21 Intellectual Property Management and Protection (IPMP) Components [7] standard specification to describe and associate this information to the multimedia content.

Moreover, the subsets defined for MPEG-21 REL to achieve interoperability in the mobile domain and presented in this paper could also be considered as mobile profiles for MPEG-21 REL.

2. Rights Expression Languages

Digital Right Management (DRM) needs technologies to protect and securely deliver digital content. To achieve this, it is also needed to have a Rights Expression Language (REL), that is a formal language used to specify this protection and secure

delivery. A REL is a formal language, designed to express rights and conditions for digital content access.

A Rights Expression Language can be used for example to control the number of times that a right is exercised over a certain digital content, express the copyright associated to a given digital content, describe an agreement between a content provider and a distributor, or between a distributor and an end user, etc.

Several RELs have been proposed to describe licenses governing the terms and conditions of content access. In this field, the Open Digital Rights Language (ODRL) proposed by Renato Ianella and MPEG-21 REL based on the eXtensible rights Markup Language (XrML) [8] cover a prominent role. Both languages are powerful yet complex. MPEG-21 REL and ODRL are syntactically based on XML while structurally they both conform to the axiomatic principles of rights modelling first laid down in the Digital Property Rights Language (DPRL) [9].

2.1 ODRL

The ODRL is a proposed language for the DRM community for the standardisation of expressing rights information over content. The ODRL is intended to provide flexible and interoperable mechanisms to support transparent and innovative use of digital resources in publishing, distributing and consuming of electronic publications, music, audio, movies, digital images, learning objects, computer software and other creations in digital form. This is an XML-based usage grammar.

Using ODRL it is possible to specify, for a digital resource (music work, content, service, or software application), which is allowed to use that resource, the rights available to them and the terms, conditions or restrictions necessary to exercise those rights on the resource. The ODRL function is to express rights granted by some parties for specific resources and the conditions under which those rights apply.

ODRL is based on an extensible model for rights expressions, which involves three core entities and their relationships in a DRM license (see Figure 1):

- **Party** includes end users and Rights Holders. Party can be an entity such as the person, organisation, or device to whom rights are granted.
- **Right** includes permissions, which can then contain constraints, requirements, and conditions. Permissions are the actual usages or activities allowed over the assets (e.g. play, print, etc.) Constraints are limits to these permissions (e.g. print an e-book for a maximum of 3 times) Requirements are the obligations needed to exercise the permission. Conditions specify exceptions that, if they become true, expire the permissions and re-negotiation may be required.
- **Asset** includes any physical or digital content. They must be uniquely identified and may consist of many subparts and be in many different formats. Assets can also be non-tangible expressions of works and/or manifested in particular renditions.

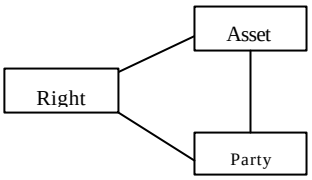


Figure 1. Core elements of ODRL

ODRL includes a data dictionary, which is formed by elements that defines permissions, rights, constraints, and requirements used in an ODRL license. All these elements form the basis of the language and can be extended by additional new elements.

For example, consider an e-book distributed to a consumer (Alice) that she can print 3 times. The ODRL license has a sentence that says that Alice is granted with the right to print the book for 3 times. In this case, Alice is a party, the book is an asset, print is a right, and "3 times" is a constraint included in the right element. Figure 2 shows this example.

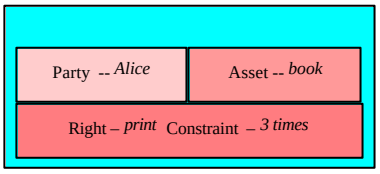


Figure 2. ODRL core elements example.

2.2 MPEG-21 REL

The REL from MPEG-21 is based on the XrML proposal. Using MPEG-21 REL it is possible to specify, for a digital resource (content, service, or software application), who is allowed to use that resource, the rights available to them and the terms, conditions or restrictions necessary to exercise those rights on the resource.

Part 5 of the MPEG-21 standard specifies the syntax and semantics of a Rights Expression Language.

MPEG-21 Rights Expression Language (REL) specifies the syntax and semantics of the language for issuing rights for Users to act on Digital Items, their Components, Fragments, and Containers.

MPEG-21 REL makes use of the Rights Data Dictionary [10], part 6 of the MPEG-21 standard, that comprises a set of clear, consistent, structured, integrated and uniquely identified terms. The structure of the RDD is designed to provide a set of well-defined terms for use in rights expressions.

At the heart of REL is the REL Core Schema whose elements and types define the core structural and validation semantics that comprises the essence of the specification. The REL Core Schema includes Core Principals, Core Rights, Core Resources and Core Conditions.

The core data model is enhanced by a number of so-called "Extensions" which add both functionality and applicability.

The most important concept in REL is the license that conceptually is a container of grants, each one of which conveys to a principal the sanction to exercise a right against a resource. The structure of a license is shown in Figure 3.

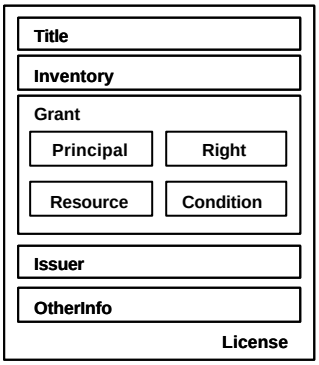


Figure 3. MPEG-21 REL license

The Title element provides a descriptive phrase about the License that is intended for human consumption in user interfaces. The Inventory element is used for defining variables within a License. The grant or grantGroup element expresses an assertion that some Principal may exercise some Right against some Resource, subject, possibly, to some Condition.

The grant or grantGroup is formed by the following four elements (see Figure 4):

- **Principal:** identifies an entity such as the person, organisation, or device to whom rights are granted. Each principal identifies exactly one party. Typically, this information has an associated authentication mechanism by which the principal can prove its identity.
- **Right**: specifies the activity or action that a principal can be granted to exercise against some resource.
- **Resource**: identifies an object which the principal can be granted a right. It can be a digital work, a service or a piece of information that can be owned by a principal. A Uniform Resource Identifier (URI) can be used to identify a resource.
- **Condition**: specifies one or more conditions that must be met before the right can be exercised. For example, a principal may need to pay a fee to exercise a right, a limit to the number of times, a time interval within which a right can be exercised, etc.

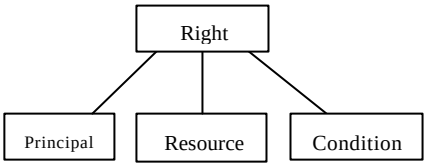


Figure 4. MPEG-21 REL core elements example.

The issuer element that represents the entity that issues the license may contain two pieces of information, a set of issuer-specific details about the circumstances under which he issues the license, and an identification of the issuer, possibly coupled with a digital signature for the license.

Finally, within the other information element, license issuers may place additional content as they find appropriate and convenient.

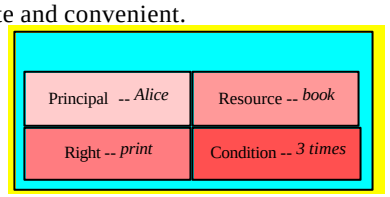


Figure 5. MPEG-21 REL example.

For example, we can consider the previous ODRL example, where an e-book is distributed to a consumer (Alice) that she can print 3 times. The MPEG-21 REL license has a sentence that says that Alice is granted with the right to print the book for 3 times. In this case, Alice is a principal, the book is a resource, print is a right, and "3 times" is a condition. In MPEG-21 REL the right-granting portion of this statement is called a grant and the entire statement is called a license. Figure 5 shows this example.

3. Intellectual Property Management and Protection

Currently, there is a lack of IPMP solutions to provide interoperability between devices and providers of content and services. Because of this fact, MPEG-21 is trying to provide a framework for the creation of new services that can be used to support new business models and that meet the needs of all members of the value chain. MPEG-21 IPMP has a very important role in the creation of these business models and must provide much more functionality than simply focusing on the content protection.

The MPEG-21 IPMP Components standard (Part 4) specifies components for IPMP applied to Digital Items to facilitate the exchange of governed and/or protected content between Peers. The MPEG-21 IPMP Components standard specifies the IPMP Digital Item Declaration Language (DIDL) that encapsulates and protects content, for example a DIDL document or part(s) thereof or asset(s), and associates appropriate identification and protection information with it. DIDL documents are specified in Part 2 of the MPEG-21 standard, Digital Item Declaration (DID), that provides an interoperable schema for declaring digital representation of works.

Moreover, MPEG-21 IPMP Components also describes, in a standardised way, information related to the IPMP Tools that protect the associated Contents, and to the licenses that govern them. The standardised IPMP info schema provides a "framework-level" description for IPMP information related to tools that protect resources or assets. It also addresses authentication of IPMP tools, and integrates rights expressions according to the Rights Data Dictionary and the Rights Expression Language.

4. OMA DRM REL and MPEG-21 REL

The Open Mobile Alliance (OMA) was formed in June 2002 by nearly 200 companies including the world's leading mobile operators, device and network suppliers, information technology companies and content and service providers. OMA specifications are the result of continuous work to define industry-wide interoperable mechanisms for developing applications and services that are deployed over wireless communication networks.

OMA DRM defines a DRM system to enable the consumption of digital content in a controlled manner, taking into account the special requirements and characteristics of the mobile domain. OMA DRM REL is defined as a mobile profile or subset of ODRL v1.1, and specifies the rights expression language used to describe mechanisms for expressing rights over DRM Content in an OMA DRM system.

There are two different versions of OMA DRM REL specification: OMA DRM REL specification v1.0 and OMA DRM REL specification v.2.0. Both specifications are defined with a Document Type Definition (DTD).

Security constitutes an important part of a DRM System, and OMA DRM REL v1.0 and, in a deeper way OMA DRM REL v2.0, provide the specification of the elements that are needed to get confidentiality , other security features, new rights and conditions.

4.1. OMA - based MPEG-21 REL v1.0

In this section, we propose an equivalent structure of the Rights Expression Language of OMA DRM REL v1.0, but defined as a subset of MPEG-21 REL, and not as a subset of ODRL (see figure 6).

The specification of OMA-based MPEG-21 REL v1.0. is defined with a DTD, and it could be considered a basic subset of OMA - based MPEG-21 REL v2.0. explained in the next section.

```
<!ELEMENT r:license ( (r:grantgroup|r:grant), r:otherinfo? )>
<!ELEMENT r:grantgroup (r:grant+)>
<!ELEMENT
  r:grant  ((mx:play|mx:execute|mx:print)?,
      r:digitalResource,  r:allConditions?)>

<!ELEMENT mx:play EMPTY>
<!ELEMENT mx:execute EMPTY>
<!ELEMENT mx:print EMPTY>

<!ELEMENT r:digitalResource (r:nonSecureIndirect) >
<!ELEMENT r:nonSecureIndirect EMPTY>
<!ATTLIST r:nonSecureIndirect URI CDATA #IMPLIED>

<!ELEMENT
       r:allConditions (sx:exerciseLimit?,
         validityInterval?, alidityIntervalDurationPattern?)>
<!ELEMENT sx:exerciseLimit (sx:count)>
<!ELEMENT sx:count (#PCDATA)>
<!ELEMENT r:validityInterval (r:notBefore?, r:notAfter?)>
<!ELEMENT r:notBefore (#PCDATA)>
<!ELEMENT r:notAfter (#PCDATA)>
<!ELEMENT sx:validityIntervalDurationPattern
(sx:duration)>
<!ELEMENT sx:duration (#PCDATA)>
<!ELEMENT r:otherinfo (version?,KeyValue?)>
<!ELEMENT version (#PCDATA)>
<!ELEMENT KeyValue (#PCDATA)>
```

Figure 6. OMA-based MPEG-21 REL DTD v1.0

4.2. OMA - based MPEG-21 REL DTD v2.0

In this section, we propose an equivalent structure of the Rights Expression Language of OMA DRM REL v2.0, but defined as a subset of MPEG-21 REL, and not as a subset of ODRL (see figures 7 and 8).

The specification of OMA-based MPEG-21 REL v2.0. is defined with a DTD, and it could be considered an extension of OMA - based MPEG-21 REL v1.0. explained in the previous section. This specification adds some relevant complexity over version 1.0. The main differences with respect to v1.0. are: inheritance model, new concepts to the security model that were missing in previous version and a data dictionary which is the result of a join of ODRL XML schema, ODRL Data Dictionary and a new OMA Data Dictionary.

```
<!ELEMENT r:license ((r:grantgroup | r:grant),
r:otherinfo?)>
<!ELEMENT r:grantgroup (r:grant+)>
<!ELEMENT r:grant
(r:keyHolder?,(mx:play| mx:execute | mx:print | inherit |
mx:move | mx:adapt | mx:execute)?, r:digitalResource,
r:allConditions?, mx:prohibitedAttributeChanges?,
r:keyHolder?)>
<!ATTLIST r:grant
        licensePartId CDATA #IMPLIED>
<!ELEMENT r:keyHolder (r:info)>
<!ELEMENT r:info (version?, uid?)>
<!ELEMENT mx:play EMPTY>
<!ELEMENT mx:execute EMPTY>
<!ELEMENT mx:print EMPTY>
<!ELEMENT mx:move EMPTY>
<!ELEMENT mx:adapt EMPTY>
<!ELEMENT inherit EMPTY>
<!ATTLIST inherit
        URI CDATA #IMPLIED>
<!ELEMENT mx:prohibitedAttributeChanges (set+)>
<!ELEMENT set EMPTY>
<!ATTLIST set
        definition CDATA #REQUIRED>
<!ELEMENT r:digitalResource (r:nonSecureIndirect)>
<!ELEMENT r:nonSecureIndirect EMPTY>
<!ATTLIST r:nonSecureIndirect
        URI CDATA #IMPLIED>
<!ELEMENT r:allConditions (sx:exerciseLimit?,
r:validityInterval?, sx:validityIntervalDurationPattern?,
mx:destination?, mx:validityTimeMetered?, mx:renderer?)>

<!ELEMENT mx:destination (r:keyHolder)>
<!ELEMENT mx:renderer (r:keyHolder)>
<!ELEMENT sx:exerciseLimit (sx:count)>
<!ELEMENT sx:count (#PCDATA)>
<!ELEMENT r:validityInterval (r:notBefore?,
r:notAfter?)>
```

Figure 7. OMA-based MPEG-21 REL DTD v2.0 part 1

```
<!ELEMENT r:otherinfo (version?, KeyValue?, uid*,
exerciseLimitDuration?, ((digest?, KeyInfo?) |
grantgroup))?>

<!ELEMENT version (#PCDATA)>

<!ELEMENT exerciseLimitDuration (#PCDATA)>
<!ELEMENT keyValue (#PCDATA)>
<!ELEMENT uid (#PCDATA)>

<!ELEMENT grantgroup (grant+)>
<!ELEMENT grant (digest?, KeyInfo?)>
<!ATTLIST grant
        licensePartIdRef CDATA #IMPLIED
>

<!ELEMENT digest (dsig:DigestMethod, dsig:DigestValue)>

<!ELEMENT dsig:DigestMethod (#PCDATA)>
<!ATTLIST dsig:DigestMethod
        Algorithm CDATA #FIXED
"http://www.w3.org/2000/09/xmldsig#sha1"
>

<!ELEMENT dsig:DigestValue (#PCDATA)>
<!ELEMENT KeyInfo (xenc:EncryptedKey?,
ds:RetrievalMethod?)>
<!ELEMENT xenc:EncryptedKey (ds:KeyInfo?,
xenc:EncryptionMethod, xenc:CipherData)>
<!ELEMENT xenc:EncryptionMethod (#PCDATA)>
<!ATTLIST xenc:EncryptionMethod
        Algorithm CDATA #FIXED
"http://www.w3.org/2001/04/xmlenc#kw-aes128"
>
<!ELEMENT xenc:CipherData (xenc:CipherValue)>
```

Figure 8. OMA-based MPEG-21 REL DTD v2.0 part 2

5. Interoperability between mobile profiles

As said before, ODRL and MPEG-21 REL have many syntactically an structurally similarities.

These RELs are widely used, so it is very important to permit interoperability between different systems that use them. They have the same objective and they start from the same base.

To transform an ODRL license into an MPEG-21 REL license, or vice versa, is equivalent to transform a XML document to another XML document, where the information they represent is the same one, but with a different XML structure.

This is only possible when both specifications convey the same semantics, as it is the case in our mapping from OMA DRM REL to MPEG-21 REL.

In order to obtain this transformation, XSL (Extensible Stylesheet Language) can be used [1]. XSLT applies transformation rules to the document source and, by changing the tree structure, produces a new document, such as another XML document. It can also amalgamate several documents into one, or even produce several documents starting from the same XML document.

If we consider the similarities between both languages, and the similarities between the previous equivalent DTDs (those shown in sections 4.1 and 4.2), it can be concluded that the interoperability between both languages is possible only for specific profiles, in this case mobile profiles. To achieve this interoperability or syntactical transformation, XSLT can be used.

6. Interoperability between MPEG-21 REL and OMA DRM REL v1.0

This section contains a table with the XML equivalences between the OMA ODRL profile and the OMA-based MPEG-21 REL subset, previously defined (see section 4.1). These equivalences will lead us to achieve interoperability between this MPEG-21 REL subset for the mobile domain and OMA DRM REL specification, doing a XSLT transformation.

OMA DRM REL v1.0 is fully supported by OMA DRM REL v2.0, specification that contains the version 1.0. A more precise explanation about REL elements used in the table 1 are given in the next section about version 2.

OMA ODRL	**OMA-based MPEG-21 REL**
<o-ex:rights>	<r:license>
<o-ex:context> <o-dd:version>1.0 </o-dd:version> </o-ex:context>	<r:otherInfo> <version>1.0 </version> </r:otherInfo>
<o-ex:asset> <o-ex:context> <o-dd:uid> cid:4567829547@foo.com </o-dd:uid> </o-ex:context> </o-ex:asset>	<r:digitalResource> <r:nonSecureIndirect URI='cid:4567829547@foo.com' /> </r:digitalResource>
<o-dd:display/>	<mx:play />
<o-dd:play/>	<mx:play />
<o-ex:permission> <o-dd:display> <o-ex:constraint> <o-dd:count> 1 </o-dd:count>	<r:allConditions> <sx:exerciseLimit> <sx:count>1</sx:count> </sx:exerciseLimit> </r:allConditions>
</o-ex:constraint> </o-dd:display> </o-ex:permission>	
<o-ex:asset> <o-ex:context> <o-dd:uid> cid:4567829547@foo.com </o-dd:uid> </o-ex:context> <ds:KeyInfo> <ds:KeyValue> vUEwr8LzEJoeiC+dgT1mgg== </ds:KeyValue> </ds:KeyInfo> </o-ex:asset>	<r:digitalResource> <r:nonSecureIndirect URI='cid:4567829547@foo.com' /> </r:digitalResource> ... <r:otherInfo> <KeyValue> vUEwr8LzEJoeiC+dgT1mgg== </KeyValue> </r:otherInfo>

Table 1. XML equivalences

7. Interoperability between MPEG-21 REL and OMA DRM REL v2.0

In this section, we introduce different tables with XML equivalences, between the OMA DRM REL v2.0 and the related MPEG-21 REL subset (see section 4.2). These equivalences will lead us to achieve also interoperability with XSLT transformation between this second MPEG-21 REL subset for the mobile domain and OMA DRM REL specification.

Different models are used to group the XML equivalences according to their functionality and license structure. The models used in this section are: Basic equivalences, Rights, Conditions, Security information association, Security and Inherit model.

In the first four models we are defining the elements that form the subset of MPEG-21 REL that fulfils OMA DRM REL v2.0 specification. The security model can be mapped to MPEG-21 REL by defining the corresponding elements in MPEG-21 <otherinfo> element or using the MPEG-21 IPMP Components specification. Finally, MPEG-21 has been extended to represent the OMA DRM <inherit> element.

7.1. Basic

The basic equivalences (see Table 2) constitutes the basis for licences and includes the necessary elements in any license. The OMA DRM REL <rights> and <asset> elements are represented with the MPEG 21 REL <license> and <digitalResource> elements. The OMA <context> element provides meta information about the rights, and is represented with the MPEG-21 <otherinfo> element.

OMA DRM REL v2.0	OMA-based MPEG-21 REL
<o-ex:rights>	<r:license>
<o-ex:context> <o-dd:version>2.0</o-dd:version> <o-dd:uid>RightsObjectID</o-dd:uid> </o-ex:context>	<r:otherinfo> <version>1.0 </version> <uid>RightsObjectID</uid> </r:otherinfo>
<o-ex:asset> <o-ex:context> <o-dd:uid>ContentID</o-dd:uid> </o-ex:context> </o-ex:asset>	<r:digitalResource> <r:nonSecureIndirect URI='ContentID' /> </r:digitalResource>

Table 2. Basic equivalences

7.2. Rights

This table 3 introduces the MPEG-21 REL rights equivalent to the rights specified in OMA DRM REL. The <display> and <play> elements are represented with the <play> element, the <export - move> element with the <move> element and the <export - copy> element with the <adapt> element and <prohibitedAttributeChanges> elements.

OMA DRM REL v2.0	OMA-based MPEG-21 REL
<o-dd:display/>	<mx:play />
<o-dd:play/>	<mx:play />
<o-dd:execute/>	<mx:execute />
<o-dd:print/>	<mx:print />
<oma-dd:export oma-dd:mode="move"> <o-ex:constraint> <oma-dd:system> <o-ex:context> <o-dd:version> 1.0 </o-dd:version> <o-dd:uid> XYZ </o-dd:uid> </o-ex:context> </oma-dd:system> </o-ex:constraint> </oma-dd:export>	<mx:move/> <r:digitalResource> <r:nonSecureIndirect URI="ContentID"/> </r:digitalResource> <r:allConditions> <mx:destination> <r:keyHolder> <r:info> <version>1.0</version> <uid>XYZ</uid> </r:info> </r:keyHolder> </mx:destination> </r:allConditions>
<oma-dd:export oma-dd:mode="copy"> <o-ex:constraint> <oma-dd:system> <o-ex:context> <o-dd:version> 1.0 </o-dd:version>	<mx:adapt/> <r:digitalResource> <r:nonSecureIndirect URI="ContentID1"/> </r:digitalResource> <mx:prohibitedAttributeChanges> <set definition=
<o-dd:uid> XYZ </o-dd:uid> </o-ex:context> </oma-dd:system> </o-ex:constraint> </oma-dd:export>	"urn:mpeg:mpeg21:2003:01-RDD-NS:2346"/> <set definition= "urn:mpeg:mpeg21:2003:01-RDD-NS:2347"/> </mx:prohibitedAttributeChanges> <r:keyHolder> <version>1.0</version> <uid>XYZ</uid> </r:keyHolder>

Table 3. Rights

7.3.Time conditions

This table 4 introduces the MPEG-21 REL time conditions equivalent to the ones specified in OMA DRM REL. The <datetime> element represented in MPEG-21 REL with the <validityInterval> element specifies an interval of time within which a right can be exercised. The <interval> represented in MPEG-21 REL with the <validityIntervalDurationPattern> element specifies a period of time within which a right can be exercised. Finally, the <accumulated> element represented in MPEG-21 REL with the <validityTimeMetered> specifies the maximum period of metered usage time during which the rights can be exercised.

OMA DRM REL v2.0	OMA-based MPEG-21 REL
<o-ex:constraint> <o-dd:datetime> <o-dd:start>... </o-dd:start> <o-dd:end>... </o-dd:end> </o-dd:datetime> </o-ex:constraint>	<r:allConditions> <r:validityInterval> <r:notBefore>...</r:notBefore> <r:notAfter>...</r:notAfter> </r:validityInterval> </r:allConditions>
<o-ex:constraint> <o-dd:interval> </o-dd:interval> </o-ex:constraint>	<r:allConditions> <sx:validityIntervalDurationPattern> <sx:duration> </sx:duration> </sx:validityIntervalDurationPattern> </r:allConditions>
<o-ex:constraint> <o-dd:accumulated> PT10H </ o-dd:accumulated> </o-ex:constraint>	<r:allConditions> <sx:validityTimeMetered> <sx:duration>PT10H</sx:duration> </sx:validityTimeMetered> </r:allConditions>

Table 4. Time conditions

7.4. More conditions

In the table 5 we introduce the rest of MPEG-21 REL conditions considered in the mobile subset we are

defining equivalent to the ones specified in OMA DRM REL. The <count> element represented in MPEG-21 REL with the <exerciseLimit> element specifies the number of allowed exercises. The <timed-count> element specify the number of times a permission may be granted over an asset or resource, with the addition of an optional timer attribute. This timer attribute specifies the number of seconds after which the count state can be reduced. As the timer attribute is not specified in MPEG-21 REL, we have defined the <exerciseLimitTime>, that consist of <count> and <duration> elements. The <individual> represented in MPEG-21 REL with the <keyHolder> element specifies the individual to which content is bound. The <system> represented in MPEG-21 REL with the <renderer> element specifies the target system to which DRM Content and Rights Objects can be exported.

OMA DRM REL v2.0	OMA-based MPEG-21 REL
<o-ex:constraint> <o-dd:count > 1 </o-dd count> </o-ex:constraint>	<sx:exerciseLimit> <sx:count>1</sx:count> </sx:exerciseLimit>
<o-ex:constraint> <o-dd:timed-count timer="30">1 </o-dd:timed-count> </o-ex:constraint>	<r:otherinfo> <exerciseLimitTime> <sx:count>1</sx:count> <sx:duration>30 </ sx:duration> </exerciseLimit> </r :otherinfo>
	<r:grant licensePartId="Asset-1"> <r:allConditions> <sx:exerciseLimit> <sx:count>1</sx:count> </sx:exerciseLimit> </r:allConditions> </r:grant licensePartId="Asset-1"> <r:otherinfo> <grant licensePartIdRef="Asset-1"> <exerciseLimitDuration> 30 </exerciseLimitDuration> </grant> </r:otherinfo>
	<sx:exerciseLimit> <r:serviceReference licensePartIdRef="externalService"/> <sx:count>1</sx:count> </sx:exerciseLimit>
<o-ex:constraint> <o-dd:individual> <o-ex:context> <odd:uid> XYZ </odd:uid> </o-ex:context> </o-dd: individual> </o-ex:constraint>	<r:grant> <r:keyHolder> <r:info> <uid>XYZ</uid> </r:info> </r:keyHolder> </r:grant>
<o-ex:constraint> <oma-dd:system>	<mx:renderer> <r:keyHolder>
<o-ex:context> <odd:uid> XYZ </odd:uid> </o-ex:context> </oma-dd system> </o-ex:constraint>	<r:info> <uid>XYZ</uid> </r:info> </r:keyHolder> </mx:renderer>

Table 5. More conditions

7.5. Security

Security constitutes an important part of a DRM system. OMA DRM REL v 2.0 provides confidentiality for the CEK (Content Encryption Key) of Rights Objects, integrity of the association between Rights Objects and DRM Content and Rights Object integrity and authenticity.

In MPEG-21 REL the security issue is not considered. To provide the OMA DRM REL security functionalities in MPEG-21 REL, we have considered two approaches. The first one is to define this security information within the MPEG-21 REL <otherinfo> element, as defined in the table 6. The second one is to consider MPEG-21 IPMP Components specification, as explained in section 8.

OMA DRM REL v2.0	OMA-based MPEG-21 REL
<o-ex:agreement> <o-ex:digest> <ds:DigestMethod Algorithm="..."/> <ds:DigestValue> DCFHash </ds:DigestValue> </o-ex:digest> </o-ex:agreement>	<r:otherinfo> <digest> <dsig:DigestMethod Algorithm="..."/> <dsig:DigestValue> DCFHash </dsig:DigestValue> </digest> </r:otherinfo>
<o-ex:agreement> <ds:KeyInfo> <xenc:EncryptedKey> <xenc:EncryptionMethod Algorithm="..."/> <xenc:CipherData> <xenc:CipherValue> EncryptedCEK </xenc:CipherValue> </xenc:CipherData> </xenc:EncryptedKey> <ds:RetrievalMethod URI="REKRe"/> </ds:KeyInfo> </o-ex:agreement>	<r:otherinfo> <KeyInfo> <xenc:EncryptedKey> <xenc:EncryptionMethod Algorithm="..."/> <xenc:CipherData> <xenc:CipherValue> EncryptedCEK </xenc:CipherValue> </xenc:CipherData> </xenc:EncryptedKey> <ds:RetrievalMethod URI="REKRe"/> </KeyInfo> </r:otherinfo>

Table 6. Security

7.6. Security information association

The table 7 explains how to associate security information to different assets or resources in the same license. The MPEG-21 REL <otherinfo> element

includes a <grant> element with the security information and a reference to the grant related.

OMA DRM REL v2.0	OMA-based MPEG-21 REL
<o-ex:agreement> *<o-ex:asset o-ex:id="A-1">* *<o-ex:digest>...</o-ex:digest>* *<ds:KeyInfo>..</ds:KeyInfo>* *</o-ex:asset>* *<o-ex:asset o-ex:id="A-2">* *<o-ex:digest>...</o-ex:digest>* *<ds:KeyInfo>...</ds:KeyInfo>* *</o-ex:asset>* *</o-ex:agreement>*	*<otherinfo>* *<r:grantgroup>* *<r:grant licensePartId="A-1">* *</r:grant>* *<r:grant licensePartId="A-2">* *</r:grant>* *<r:otherinfo>* *<grant licensePartIdRef="A-1">* *<digest>...</digest>* *<KeyInfo>...</KeyInfo>* *</grant>* *<grant licensePartIdRef="A-2">* *<digest>...</digest>* *<KeyInfo>...</KeyInfo>* *</grant>* *</r:grantgroup>* *</r:otherinfo>*

Table 7. Security information association

7.7. Inherit

The OMA DRM REL inherit model is not considered in MPEG-21 REL, therefore MPEG-21 REL has been extended with a new <inherit> right, as we show in the table 8. A License called parent license defines Permissions and Constraints for DRM Content which can be inherited by a new license called child License. In the child license we only include a reference to the parent license, and then the child license inherits permissions and constraints from the parent license.

OMA DRM REL v2.0	OMA-based MPEG-21 REL
<o-ex:asset> *<o-ex:inherit>* *<o-ex:context>* *<o-dd:uid>Subs</o-dd:uid>* *</o-ex:context>* *</o-ex:inherit>* *<o-ex:asset>*	*<r:grant>* *<inherit URI="Subs"/>* *</r:grant>*

Table 8. Inherit

8. Protection of multimedia content

OMA and MPEG-21 standards have considered a different approach in the specification of protection and governance information and their association with digital content. OMA DRM REL v2.0 includes protection information within the licenses, while in the MPEG-21 standard the protection information and the mechanisms to associate it, together with licenses, to protected and governed content is defined in IPMP Components, Part 4 of the MPEG-21 standard.

To achieve interoperability between OMA and MPEG-21 standard we have considered two approaches. The first one, presented in section 7, is to define a mobile profile for MPEG-21 REL and RDD defining a subset of rights, resources and conditions according to OMA DRM REL v2.0 specification, but extending it by defining the appropriate elements (see section 7.5) for the protection information and DRM content association integrity. The second approach is to consider MPEG-21 IPMP Components specification to describe protection information and restrict MPEG-21 REL and RDD parts for mobile applications. Note that in this approach we only have to consider the extension done in MPEG-21 REL for the inheritance model as specified in section 7.7 to achieve interoperability with OMA DRM REL v2.0 specifications.

In this section we present how protection information is described and associated to content, using MPEG-21 IPMP technologies. On the other hand, the licenses are generated according to the profile defined for MPEG-21 REL in section 7 without considering the extension proposed for protection information in security section.

```
<didl:DIDL>
 <didl:Item>
  <didl:Component>
   <didl:Resource mimeType="application/ipmp">
    <ipmpdidl:ProtectedAsset mimeType="video/mpeg">
     <ipmpdidl:Info>
      <IPMPInfoDescriptor>
       <Tool>
          Protection tools information
       </Tool>
       <RightsDescriptor>
        <License>
         <r:license>
          <r:grant>
           <inherit URI="SubscriptionGUID"/>
           <r:digitalResource>
            <r:nonSecureIndirect URI="ContentID"/>
           </r:digitalResource>
          </r:grant>
          <r:otherinfo>
           <version>2.0</version>
           <uid>RightsObjectID</uid>
          </r:otherinfo>
         </r:license>
        </License>
       </RightsDescriptor>
      </ipmpdidl:Info>
     <ipmpdidl:Contents ref="ContentID"/>
    </ipmpdidl:ProtectedAsset>
   </didl:Resource>
  </didl:Component>
 </didl:Item>
</didl:DIDL>
```

Figure 9. Protected and governed asset

Figure 9 shows how protection and governance information is described and associated to digital content, using IPMP and MPEG-21 REL technologies equivalent to OMA DRM REL v2.0. Specifically, protection tools are described using the MPEG-21 IPMP Information Descriptor schema and governance information using the subset of MPEG-21 REL defined for the mobile profile. Finally, IPMP-DIDL schema is used to associate this information with the correspondent asset.

Main difference of the two approaches considered is the information expressed in the licenses. In the first one, licenses contain information related to the rights and conditions of use of digital content and content protection information, while in the second one protection information is not described within licenses. Figure 10 shows how an encryption tool is described in the second approach presented using MPEG-21 IPMP Components specification. The "Tool" element contains relevant information of the tool that will be used to decrypt the content, as its unique identifier, the remote location from where it can be retrieved, and its initialization settings where two different types of data are placed. On one hand, the "InitializationData" that contains the key (CEK) for decrypting the content using the tool previously described, this key is also encrypted (EncryptedCEK). On the other hand, the information of the tool that will be used to decrypt this key (CEK) and a reference to the key used to encrypt the CEK.

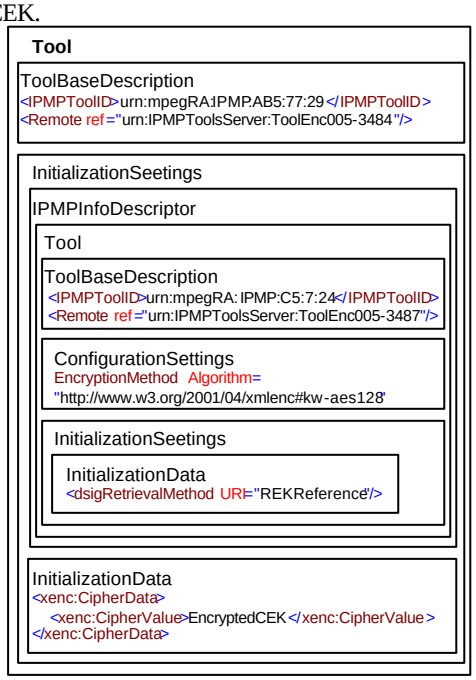


Figure 10. Encryption tool description example

9. Associated tools

In order to validate our proposal, we have adapted some of our previous tools [11], already contributed as MPEG-21 Reference Software [12], in order to work with this subset in both cases, OMA ODRL and MPEG-21 REL.

9.1. DMAG checker

The DMAG Checker (DC) is an application that syntactically validates a REL license, and subsets of them as the profiles proposed in this paper, against the DTD or XML Schema used by the license.

This software has been developed in Java. It can run on MS-Windows and Linux platforms. The parser used in the implementation is the Xerces parser. The output of the DC is a message reporting if the license is syntactically valid or not, according to the DTD or XML Schema specified within the license. If the license is not valid, the DC informs about the reasons why.

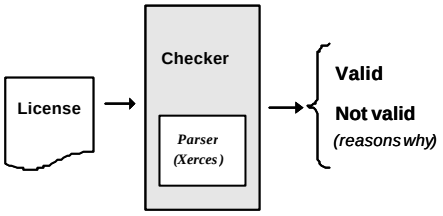


Figure 11. DMAG Checker

9.2. DMAG License Creator for mobile profiles

The DMAG License Creator for mobile profiles (subsets specified in this paper) is a software implementation that creates OMA – based MPEG-21 REL licenses equivalent to the OMA DRM v2.0 ones, and OMA DRM v2.0 licenses. This software has been developed in Java. It can run on MS-Windows and Linux platforms.

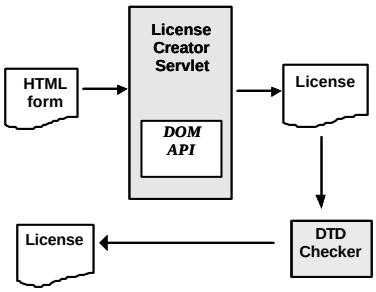


Figure 12. DMAG License Creator

9.3. DMAG License Translator

In order to implement a license translator for the mobile profile presented in this work, we have developed utilities to transform OMA – based MPEG-21 REL v1.0 and v2.0 licenses to OMA DRM REL v1.0 and v2.0 and in the reverse direction. These utilities have been developed using XSLT and permit to do a syntactical translation between the mobile profiles presented.

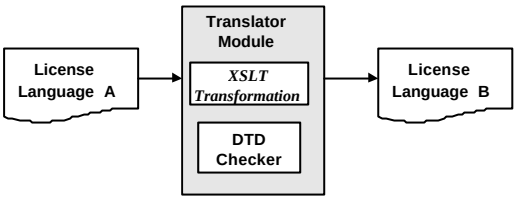


Figure 13. DMAG License Translator

10. Conclusions and further work

This paper has shown how interoperability and translation of licenses among different RELs (Rights Expression Languages) can be achieved. We have presented different possible solutions and tools to achieve interoperability among different versions of OMA DRM REL and MPEG-21 REL by defining subsets or profiles of MPEG-21 REL that provide the same functionalities as the OMA DRM REL, which is a mobile profile of ODRL. Tools to create and check licenses in a mobile profile have been introduced.

Furthermore, as the MPEG-21 REL standard specification does not provide all functionalities required by OMA, we have proposed to extend it and used MPEG-21 IPMP to fulfil security requirements.

However, the purpose of this paper has not been to define a formal mobile profile, since this should rather be an initiative from the interested industry. On the contrary, we are proposing a possible approach for the specification, and further implementation, of MPEG-21 subsets able to interoperate with other RELs.

Currently, we are working to extend the capabilities of the tools. The main objective of our future work is to expand the scope of this set of tools (generators and converters) to permit that every system could work in MPEG-21 REL or ODRL without distinction, transparently to the user.

Acknowledgements. This work has been partly supported by the Spanish administration (AgentWeb project, TIC 2002-01336) and is being developed within VISNET [13], a European Network of Excellence, and AXMEDIS [14], a European Integrated Project, both funded under the European Commission IST FP6 program.

11. References

[1] J. Polo, J. Prados and J. Delgado. Interoperability between ODRL and MPEG-21 REL. First International ODRL Workshop. Vienna (Austria). April 2004. ISBN 1-74064-500-6.

[2] Open Digital Rights Language (ODRL). http://odrl.net.

[3] ISO/IEC, ISO/IEC IS 21000-5 – Rights Expression Language.

[4] OMA, http://www.openmobilealliance.org/

[5] Rights Expression Language. Approved Version 1.0 – 15 June 2004. Open Mobile Alliance. OMA-Download-DRMREL-V1_0-20040615-A.

[6] Rights Expression Language. OMA-Download-DRMREL-V2_0-20041210-C. 10 December 2004.

[7] ISO/IEC, ISO/IEC CD 21000-4 – Intellectual Property Management and Protection.

[8] XrML, http:// www.xrml.org/.

[9] DPRL, http://www.oasis-open.org/cover/DPRLmanual-XML2.htm.

[10] ISO/IEC, "ISO/IEC FDIS 21000-6 - Rights Data Dictionary. ISO/IEC JTC 1/SC 29/WG 11/N5842", July 2003.

[11] DMAG MPEG REL reference software. http://dmag.upf.edu/DMAGRELTools/Index.htm

[12] Draft DoC on ISO/IEC 21000-8 FCD MPEG-21 Reference Software. IEC C1/SC29/WG11N6744. October 2004.

[13] Networked Audiovisual Media Technologies (VISNET), IST-2003-506946, http://www.visnet-noe.org.

[14] Automatic Production of Cross Media Content for Multi-channel Distribution (AXMEDIS), IST-2004-511299, http://www.axmedis.org.

Embedding ODRL statements in Dublin Core

Enric Peig and Jaime Delgado, *Universitat Pompeu Fabra, Barcelona, Spain*

***Abstract*— Dublin Core is a standard for creating metadata records about resources. Over these resources we can define policies of usage. ODRL is an initiative to express the rights statements over the resources, with the idea of developing tools to enforce the policies defined. The normative way to express ODRL statements is in XML syntax, which is rather difficult for a person to read and understand. The easiest way to relate the license to the Dublin Core metadata set is to include a link to the license into the metadata record. This can be useful for the automatic processing of the license but, however, doesn't give descriptive information to the user. In this paper we propose a mechanism to obtain this descriptive information, by converting the ODRL statements into textual information, and embed it in Dublin Core metadata records, in order to ease its human comprehension.**

***Index Terms*—Metadata, digital rights management.**

I. Introduction

ODRL (Open Digital Rights Language) [1] is a key tool for the digital rights management of electronic publications. It consists of a language for expressing the rights and a data dictionary that establishes the semantics of every entity defined in the ODRL Foundation Model. The normative way to express ODRL is in schema-valid XML syntax, in order to be easily processed by DRM tools.

On the other hand, Dublin Core is a standard for creating descriptive metadata records about resources. The ODRL community has realised the need of combining the ODRL rights expressions with descriptive metadata records. With this goal in mind, a joint working group between ODRL and DCMI (Dublin Core Metadata Initiative) [2] has been established to study the possibility of creating an ODRL profile that enables this combination.

Dublin Core and ODRL serve different purposes. While a metadata record following the Dublin Core standard aims to describe different characteristics about a resource, an ODRL statement is meant to provide the mechanisms to enforce a usage policy over a resource.

In this paper, we propose a mechanism to embed the rights statements expressed in ODRL beneath a metadata record associated to a resource, to which the rights statements apply, focusing on the Dublin Core Metadata standard.

This work has been partly supported by the Spanish administration (AgentWeb project, TIC 2002-01336) and is being developed within VISNET (IST-2003-506946, http://www.visnet-noe.org), a European Network of Excellence funded under the European Commission IST FP6 program.

The remainder of this paper is structured as follows. In section 2 we give an overview of the Dublin Core Metadata Element Set, focusing on the elements related to the rights description. Section 3 describes our proposal for embedding ODRL in Dublin Core, analyses the different ODRL models and discusses which of them are to be considered and which not. Then, section 4 concludes the paper.

II. The Dublin Core standard

A. Dublin Core Metadata Element Set

The Dublin Core metadata element set is a standard for information resource description. Simple Dublin Core consists of 15 descriptive semantic definitions and represents a core set of elements likely to be useful across a broad range of applications, whereas Qualified Dublin Core includes additional elements, as well as a group of element refinements (also called qualifiers) that refine the semantics of the core elements in ways that may be useful in resource discovery. Also, the usage of controlled vocabularies for some elements is encouraged, thus avoiding misspellings and confusions, and increasing interoperability.

The DCMI (Dublin Core Metadata Initiative) is the organisation who is in charge of the maintenance of the standard, the promotion of its usage and the proposal of new elements, qualifiers and encoding schemes. In this moment, it is an ISO standard (ISO 15836:2003), a NISO standard (ANSI/NISO Z39.85-2001), a CEN recommendation (CWA 13874) and an IETF RFC (RFC 2413).

The 15 core elements defined in Dublin Core are *contributor*, *coverage*, *creator*, *date*, *description*, *format*, *identifier*, *language*, *publisher*, *relation*, *rights*, *source*, *subject*, *title* and *type*.

The proposed new elements after the establishment of the core are *audience*, *provenance* and *rightsHolder*, whereas examples of qualifiers are *abstract*, which is a refinement of *description*; *created*, *dateCopyrighted* or *dateAccepted*, which are refinements of *date*; *hasPart* or *isPartOf*, which are refinements of *relation*; or *license*, which is a refinement of *rights*.

Examples of controlled vocabularies are the DDC (Dewey Decimal Classification), or LCC (Library of Congress Classification) to be used in *subject*; the RFC1766 for languages; the IMT (Internet Media Type) for *format*, etc.

Also, the DCMI has defined some controlled vocabularies, for example, the DCMI Type classification for the element *type*, or DCMI Period, that specifies the limits of a time

interval and is useful for the element *date*.

Dublin Core can be used in many ways. The DCMI emits usage guides to assist users in creating descriptive records using Dublin Core in these different ways, from the simplest one (using only some or all the 15 core elements) to a more sophisticated one (choosing some of the qualifiers already defined).

Dublin Core is intended to be used primarily for human consumption, so the values of the elements tend to be human-readable. Nevertheless, it is also possible to use Dublin Core for automatic machine processing. In fact, we have developed a system that includes automatic access and processing of Dublin Core metadata records [3].

Original XML binding of a permission:

```
<permission>
   <display/>
   <print>
      <constraint>
         <count>5</count>
      </constraint>
   </print>
</permission>
```

Equivalent human-readable metadata record:

Rights	permission to display; permission to print with constraint 5 times

Example 1. Translation of a simple permission

B. *Rights in Dublin Core*

There is one element in the core thought to be used in the specification of the rights over the resource. It is the element called *rights*, and, according to the usage guide of this element, "typically, *rights* will contain a rights management statement for the resource, or reference a service providing such information. Rights information often encompasses Intellectual Property Rights (IPR), Copyright, and various Property Rights. If the *rights* element is absent, no assumptions may be made about any rights held in or over the resource".

Later, two refinements to this element and a new element have been proposed. The refinements are *accessRights* and *license*, and the element is *rightsHolder*. The qualifier *accessRights* is defined as "information about who can access the resource or an indication of its security status", whereas *license* is "a legal document giving official permission to do something with the resource and recommended best practice is to identify the license using a URI". The new proposed element *rightsHolder* is "a person or organisation owning or managing rights over the resource".

III. EMBEDDING MECHANISM

Keeping in mind that Dublin Core is primarily for human consumption, although machine-processable, we propose to embed the ODRL statements about a resource in the element *rights*, but in a human-readable way. So we propose to translate (or perhaps more precisely, to parse) from the XML binding of the statements to natural language, so as they can be easily understood by a human consumer. This parsing can be done automatically, without human intervention, and a key aspect is that only the terms that appear in the Rights Expression Language and in the Data Dictionary should be used.

An example of this translation is shown in Example 1.

Of course, if the element *rights* is used with this purpose, it has only informative value. It can not be reliable for a machine-driven process of analysis of the Rights statements. Then, the *license* refinement can be used, including in it the URI of the XML version of the ODRL statement, so as to point to the original ODRL license and be able to process it.

The ODRL Foundation Model consists of the following three core entities:

- Assets
- Rights
- Parties

The Rights include Permissions, which can then contain Constraints, Requirements and Conditions. There can be also Offers and Agreements, which can be accepted or revoked. Most entities can support a specific Context.

As the purpose is to inform about the rights over a resource, we only have to deal with permissions. We don't need to parse neither offers nor agreements. So, we only need to focus on the models related to Permissions, which are:

- ODRL Permission Model
- ODRL Constraint Model
- ODRL Requirement Model
- ODRL Condition Model
- ODRL Rights Holder Model
- ODRL Context Model

All these models include different terms, which are defined in the ODRL Data Dictionary. So the XML-to-natural-language parser must create human-readable sentences following the semantics included in the Data Dictionary.

We propose a specific syntax for the phrases generated:

sentence; sentence; …

where sentence consists of a permission with all the constraints, requirements and conditions that apply to it. So in the textual phrase, there will be so many sentences as permissions expressed in the whole license, separated by semi-colons.

Another example is shown in Example 2, and yet another one, more complex, is shown in Example 3.

For the sake of simplicity and ease of reading, we don't need to be exhaustive. So, the parser can be tailored to translate in different levels, from the most exhaustive one (translating all the statements literally) to a lighter one (translating only the most relevant statements).

IV. Conclusion

In this paper we have presented a mechanism to embed ODRL statements about digital rights over a resource beneath a descriptive metadata record. The idea is to give information to the users about the rights over a resource but in a human-

Original XML binding of a permission with a requirement:

```
<permission>
   <play>
      <requirement>
         <peruse>
            <payment>
               <amount currency="AUD">
                  20.00
               </amount>
               <taxpercent code="GST">
                  10.0
               </taxpercent>
            </payment>
         </peruse>
      </requirement>
   </play>
</permission>
```

Equivalent human-readable metadata record:

Rights	permission to play paying AUD $20 plus 10% tax

Example 2. Translation of a permission with a requirement

Original XML binding of two permissions, one with a specific condition, and both with another condition:

```
<permission>
   <sell/>
   <play>
      <condition>
         <constraint>
            <software>X</software>
         </constraint>
      </condition>
   </play>
</permission>
<condition>
   <constraint>
      <spatial>
         <context>
            <uid>iso3166:AU</uid>
         </context>
      </spatial>
   </constraint>
</condition>
```

Equivalent human-readable metadata record:

Rights	permission to sell valid until exercised in Australia; permission to play valid until software X is used or until exercised in Australia

Example 3. Translation of a double permission, with conditions

readable way, so we propose to translate from the XML binding to natural language, using the semantics expressed in the Data Dictionary. This translation, or parsing, can be done automatically and the level of exhaustiveness can be previously defined.

We have focused our proposal in the Dublin Core Metadata standard, but the same process can be applied to any other metadata scheme that has terms intended to carry descriptive information about rights, such as LOM (Learning Objects Metadata) [4] or SMPTE 335M [5], a metadata standard for television material.

REFERENCES

[1] R. Iannella. Open Digital Rights Language (ODRL), Version 1.1. http://odrl.net, August 2002.
[2] Dublin Core Metadata Initiative, http://www.dublincore.org
[3] R. García, R. Gil, J. Delgado. Intellectual Property Rights management using a Semantic Web information system. International Conference on Ontologies, Databases and Applications of Semantics (ODBASE 2004). LNCS, vol. 3290, pp 689-704, 2004. ISBN: 3-540-23663-5.
[4] IEEE LOM, http://ltsc.ieee.org/wg12/index.html
[5] SMPTE 335M-2001. Television - Metadata Dictionary Structure. http://www.smpte.org

Using ODRL to express rights for different content usage scenarios

Carlos Serrão, Miguel Dias and Jaime Delgado

***Abstract*— The expression of rights over generic content is one of the most important functions in any DRM system [1][2]. It is impossible to conceive such a system without the possibility to define how and under which conditions content can be used by the end-user and any other user in the content lifecycle chain. ODRL [14] represents an opportunity to have rights expression richness, flexibility and at the same time openness.**

This paper addresses those characteristics in the ODRL language by providing examples on how ODRL is currently being used in several content usage scenarios, such as music download and streaming, video-surveillance data streaming and storage and remote sensing of JPEG2000 images.

This paper also makes a short reference to the OpenSDRM architecture [3][4], an open DRM system that uses ODRL as its rights expression language, providing an interoperable rights enforcing layer. This layer acts as middleware to enforce the expressed rights over the content, through the provision of the digital Wallet concept [3]. The module which implements this concept is capable of accessing the rights locally or over the network, interpret and enforce them to the requesting content applications.

***Index Terms*— ODRL, OpenSDRM, REL, Wallet, XML**

I. Introduction

The Rights Expression definition is one of the most relevant functionalities of any DRM system [1][2]. It allows the expression of rights which are associated with a particular content and with a specific user and usage. Although this is important, rights expression is only effective if it is associated with technology that can enforce such rights on the content [1][2]. This paper describes and discusses a system, based on a client-side digital Wallet that works as an intermediate layer between the final user content rendering applications and the rights expression language. This technology is also associated with the description of license templates by different License servers. This paper also provides three different scenarios where this system is being applied together with ODRL [14]. These scenarios include the electronic commerce of digital music on a portal, the streaming of video-surveillance data and the controlled access to remote sensing images in JPEG2000 format.

The first scenario refers to one of the most attractive types of content exchanged over the Internet – digital music. Although this represents an opportunity for music producers that can use a larger massive channel to reach new consumers with radically different business models, it also represents a menace due to increasing copyright infringements [4][15]. Most of these infringements to copyright are performed while exchanging music over P2P networks. This scenario has already been developed and tested, and has been deployed on a service, which is referred to as Music-4You [26]. On this scenario, ODRL was used to express the licenses that described the rights of a certain user to access the content. Although this is an interesting scenario, it is not a new one.

The second scenario focused in this paper is the storage and streaming of video-surveillance data using JPEG2000 [18][19] (in particular Motion JPEG2000). This scenario, currently under development, uses ODRL licenses to express the rights of a particular user to access to the video-surveillance data. This scenario was recently demonstrated in the WCAM European Project, under the FP6 IST framework [21].

The final scenario which this paper describes relates to the usage of ODRL to express the rights to access JPEG2000-based Earth Observation products. This scenario has also been developed and demonstrated in the HICOD2000 European Space Agency project [25].

The paper is structured as follows: in section 2, a short description of OpenSDRM, an open DRM platform, will be presented with a specific focus on how the platform generates and manages licenses [3]. In section 3, we present the technique used by a middleware layer to manage the licenses and the rights at the client-side. Section 4, tackles the Usage Scenarios: music download and streaming; video-surveillance streaming and storage; and remote sensing of JPEG2000 images. In section 5, we extract some conclusions.

II. The OpenSDRM solution

OpenSDRM is a service-oriented DRM platform [3][4], independent from the type of content, the content protection system and the implemented business model. It can be used

Carlos Serrão and Miguel Dias are with Adetti/ISCTE - Ed. ISCTE – Av. das Forças Armadas, 1600-082, Lisboa, Portugal; (e-mail: Carlos.Serrao@iscte.pt , Miguel.Dias@iscte.pt.)

Jaime Delgado is with Universitat Pompeu Fabra, Departament de Tecnologia, Pg. Circumval·lació 8, E-08003 Barcelona, Spain (e-mail: jaime.delgado@upf.edu).

with multiple communication protocols and is based on the emerging service-oriented paradigm (SOAP [13], WSDL and UDDI) approach [3], called Service Oriented Architecture (SoA). OpenSDRM (Figure 1) covers most of the content lifecycle phases: from content authoring, distribution and management of the related rights up to the final user.

The OpenSDRM platform (Figure 1) was designed having in mind concepts such as content adaptation and a wide range of business models applicability (download, super-distribution, streaming or even broadcasting). In a more technical approach, OpenSDRM is composed by a set of external actors (red circles) or systems (orange square) and a set of internal components (inside the yellow center square) [3]. The internal components are oriented towards the service they supply, and are described in more detail in the next section. From a more technical point of view, these internal components are self-descriptive, in the sense that they expose an open WSDL description of the services they provide, and any authenticated component can connect to it and use its services – DRM services. These components communicate with each other using SOAP messages [13]. The discovery and identification of services is currently being provided by a configuration server, but this service will be provided by an UDDI server. OpenSDRM makes an extensive usage of ODRL to specify and manage the rights associated to content in each of the presented scenarios [3][4].

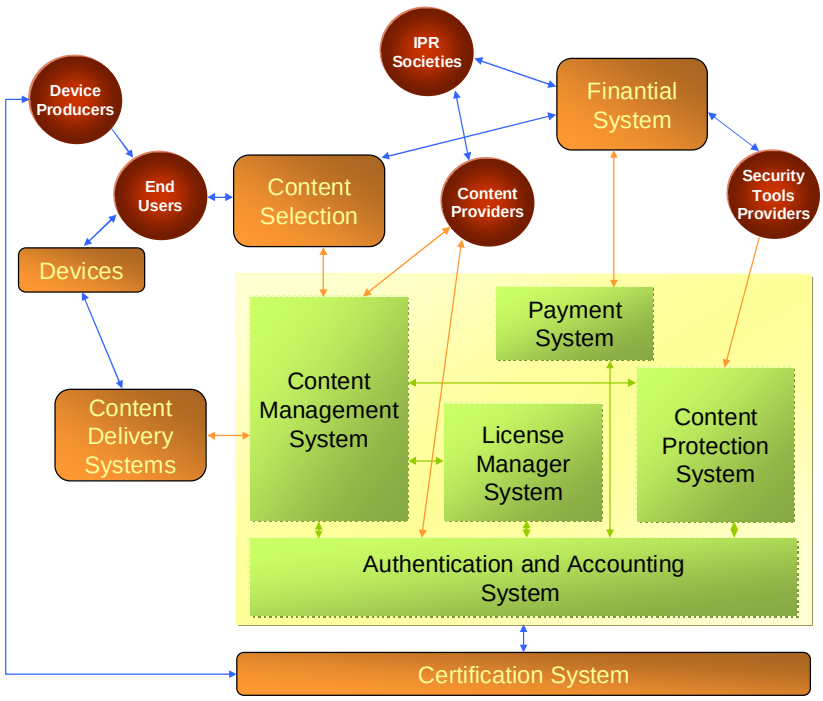


Figure 1 – OpenSDRM service-oriented architecture

A. *External Actors and Systems*

The main external actors and systems that interact with the OpenSDRM architecture (Figure 1) are: the End-Users, the Security Tools Providers, the Content Providers, the Financial System, the Content Selection System, the Content Delivery System, the Devices and the Certification System [3].

The **End-User** represents an entity who wishes to use some content. This content may or may not be protected. However, the way to access and display such content may require the use of protected devices, software and licenses. The User will make requests to OpenSDRM in order to: provide identification information, perform authentication, download licenses and use the content.

The **Security Tools Provider** is any organization that produces tools and technologies for encryption, scrambling, watermarking and others that can be applied to content protection. These tools are registered and made available to OpenSDRM for use in content rights protection. These tools will need to comply with some guidelines, defined by the platform manager. These guidelines bound together with a subscription, are translated into a business relation that must exist between a given Content Provider and the Security/Protection Tools Provider. A given producer and/or distributor of content, may want to choose which type of protection the content will have and, respectively, which tools can be applied to the content and from which supplier.

The **Content Provider** is any multimedia content supplier that feeds a Commerce Platform or a Content Management System, connected to the OpenSDRM with content and optional metadata. This information and content will be made available to End-users.

The **Financial System** facilitates the commercialization of content. OpenSDRM plays an important role since it provides the services for handling electronic payments. The interface

between OpenSDRM and the Payment Infrastructure is generic and independent from the payment method, allowing therefore a multiplicity of payment systems.

The **Content Selection System** is the module on which the End-Users can select the content that they want to enjoy. This can take the form of an Electronic Commerce site or an Electronic Program Guide.

The **Content Delivery System** is the system which is responsible for delivering the content to the End-Users or to the End-Users devices. This system is a generic entity that can be instantiated with any kind of content delivery system (download, broadcast, etc.) that is independent from the rights management system itself.

The **Device** is client-side system that represents the software or hardware that will be used to render the content. This is a generic system with the particularity of being able to display/playback the appropriate content for which the necessary audio/video codec should be available (if this codec is not available it must be downloaded from a remote secure server).

The **Certification System** is responsible for receiving requests for and issuing credentials to entities. These credentials will be used by entities to authenticate themselves to each other, allowing the establishment of secure and authenticated communication channels between them (this is part of the establishment of one of the two OpenSDRM's security layers). All the components in the OpenSDRM architecture communicate using the channel security provided by the SSL/TLS protocol [3]. This Certification System may be internal to OpenSDRM, and therefore entirely managed by some entity, or it may be an external commercial entity, such as Verisign or Thawte [3][4].

B. Internal Components & Interfaces

The main internal components of the OpenSDRM platform are: Content Management System, License Manager System, Payment System, Content Protection System and the Authentication and Accounting System [3].

The **Content Management System** is a system responsible for performing several functions. This system is responsible for content preparation and protection, content registration, content selection and trading and content delivery.

- Content preparation and protection: it receives raw content from a specified source or sources and encodes it on a specified format, adds metadata and protects it. It is not implemented using the WS approach, although it uses some components that provide such approach.
- Content registration: a function which role is to assign unique identifiers to content and to register metadata information for that specific content. The service assigns unique identifiers to content using the MPEG-21 [6] directives about Digital Item Identification (DII) [7], using a reduced version of the MPEG-21 DII Digital Object Identifiers [6][7][16].
- Content selection and trading: is an integration function responsible for establishing the liaison between the platform that actually supplies the content and the DRM platform. Normally, content is chosen via web browser, some very generic metadata might be consulted, information about the price is also available, and especially the content usage conditions might be established.
- Content delivery: is a function responsible for notifying the appropriate content servers that a given content has been requested and that needs to be feed to the final user.

The **License manager System** is a system responsible for house-keeping the rules associating a user, the content and his/her corresponding access rights. This component will accept connections from authenticated content rendering application clients for downloading licenses, which will be applied to the protected content through an appropriate protection tool. The licenses are XML formatted using Open Digital Rights Language (ODRL).

The **Payment System** is a system responsible for verifying and validating the payment methods provided by the User to the Content Management System while acquiring content.

The **Content Protection System** is the system responsible for registering new protection tools and for receiving authenticated client content rendering application requests for the downloading of a specific protection tool. It is also responsible for making protection tools available to the Content Preparation service to allow the protection of content.

The **Authentication and Accounting System** is a key-system. It is responsible for authenticating all the internal services and components as well as some external actors to the DRM system. It validates the access rights of all of them working as a single sign-on point, registering and managing components and users on the system. It uses cryptographic XML credentials to authenticate both components and users in order to authenticate the transactions exchanged between them (XML Encryption and XML Signature) [10][11].

All the above systems are interconnected and they were developed using a web-services paradigm: SOAP (Simple Object Access Protocol) and WSDL (Web Services Description Language). Each of these services is self-explanatory in terms of describing its external interfaces which allow the entrance of new components in a simple and seamless way. On the other end, each of this identified components exchange their messages, recurring to the SOAP protocol.

III. LICENSE MANAGEMENT ON THE SYSTEM

One of the more interesting mechanisms that are described on this paper relates to the fact that licenses are handled at the client-side by a middleware layer, called OpenSDRM Wallet [3][4]. This Wallet (Figure 2) is capable of managing

the access to protected content by different content handling applications. Every time an application wishes to perform an operation over the content, it contacts the Wallet that authorizes or not such operation according to what is specified on the license. This layer allows the coexistence of many DRM-protected files and DRM-enabled applications on a single client system, presenting a horizontal approach to DRM.

The OpenSDRM Wallet (Figure 2) is at the same time a Windows component which is responsible for holding down some of the user private information, such as some authentication credentials which allow the user to perform electronic payments to support the acquired content. The Wallet can store information in a secure way, either locally in the final user's computer (on an encrypted file-system or on the registry) or remotely on a server (on an encrypted database).

Figure 2 – OpenSDRM Wallet running

Nowadays, most of the existing DRM approaches are essentially vertical: examples of these include Microsoft Windows Media Rights Management (WMRM) [5][8] or Apple iTunes [9]. While a solution like Microsoft WMRM is a Microsoft end-to-end system-dependent (even at the client-side) relying on Windows Media Player to obtain the licenses and enforce them on the content [5][8][12], OpenSDRM follows a more horizontal approach in which several content applications can share the access to content, mediated by the OpenSDRM Wallet. This fact provides an important client-side interoperability layer. At the same time this approach also provides server-side interoperability since clients are independent from the server where they obtain the licenses.

A previous and important step is executed between the content application and the OpenSDRM Wallet in order to authenticate the application so that it can request content operations to the Wallet (this may include receiving content deciphering keys provided in the licenses). This means that any of the applications that wish to use this system will need to know how to execute an enrolment process composed by the following two steps:

- Enroll and request authentication to the OpenSDRM Wallet, exchanging a set of credentials with it, to enable application authentication and the establishment of a secure channel between the application and the Wallet – this secure channel will be unique by for the application and the Wallet;
- Request authorization to the OpenSDRM Wallet to perform operations over the content. This process includes the extraction of content unique identifier and requesting the Wallet the permission to use the content. The Wallet is responsible for getting the license from the server, parsing it; analyzing the rights that are associated to it before giving permission or rejecting the operation over the content (this may include passing the decryption key to the application or the appropriate protection tool).

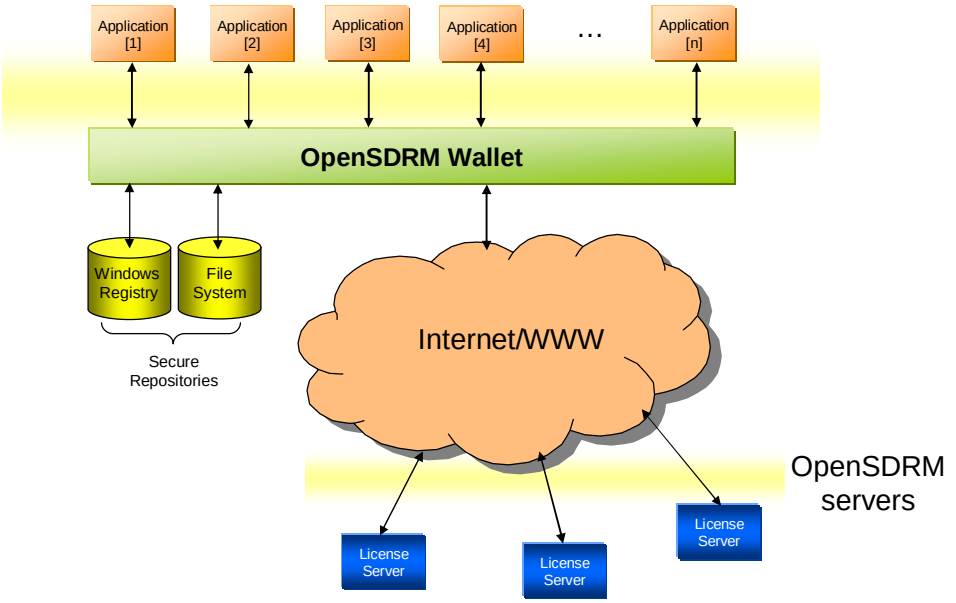


Figure 3 - OpenSDRM Wallet mediating Access to licenses

In order for this to work, the client content application does not need to know anything about the Rights Expression Language (REL) that is being used to express the rights. Therefore, this simplifies these applications design, and more, provides a layer of interoperability of the different RELs that may be used to express the rights.

Nonetheless, the content rendering application will need to be able to perform the following operations (already sketched before):

- Establish a trust relation with the underlying OpenSDRM Wallet, during a specific enrolment mechanism through the exchange of cryptographic credentials;
- Define and use a simple transaction protocol with the Wallet to request access to content operations. This transaction protocol is based on requests from the application and answers from the Wallet. An example could be a music player application asking permission to the Wallet to play a protected music track once. In this case the music player sends a message to the Wallet: “RENDER CID1234”. The Wallet receives this message and verifies that the User has a valid license. If the evaluation process is positive then the Wallet returns the key to render the content: KEY;
- Implement the necessary mechanism to establish a link with the content protection technology to be able to render it.

On the server-side of the OpenSDRM solution, one or more License servers can issue licenses (ODRL formatted, for example) [14] that are bound to the user and content. These licenses specify how content can be used by the user, according to a set of pre-established parameters.

The system contemplates the existence of one or more License Servers on the system, and also the possibility that each of the License Servers can issue more than one license type. In many current real cases, the License Server is strongly linked with the place where the content is obtained and with the implemented business model. This situation creates most of the times an unnecessary complexity in licenses issuance and management, and also trends to work as an interoperability blocking force. However, our approach tries to minimize this problem.

The foreseen OpenSDRM model is the one in which many content supply services can exist with business relationships with multiple License servers. These License servers can issue multiple licenses to many users – it is a many to many relationship.

With this idea in mind, OpenSDRM uses a template system for license creation [3]. This system allows the definition of the business model (or models) for each content business by the definition of the specific parameters that can be modified on a pre-created ODRL license template (Listing 1).

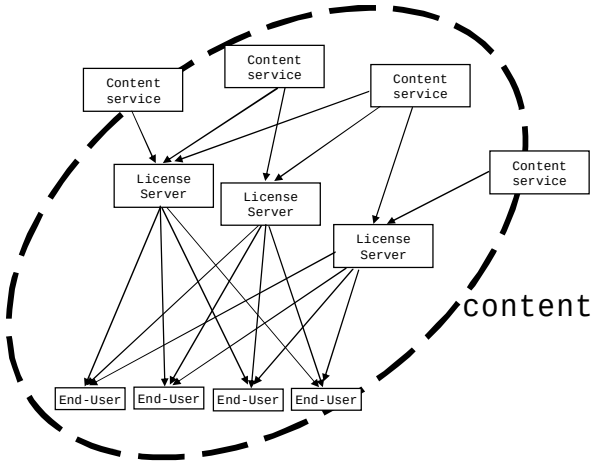


Figure 4 - License distribution schema

There can be as many templates as there are business models (or variations) on the system, and the final license is an instantiation of the business model for an end-user concerning a specific content. The following example represents an ODRL simplified license template adapted to a specific content and business model (for simplification, the presented license template does not have the Content Encryption Key (CEK) ciphered nor is digitally signed).

```
<?xml version="1.0" encoding="UTF-8" ?>
<o-ex:rights xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xmlns:o-dd="http://odrl.net/1.1/ODRL-DD"
         xmlns:ds="http://odrl.net/1.1/ODRL-DD"
         xsi:schemaLocation="http://odrl.net/1.1/ODRL-EX
../schemas/ODRL-EX-11.xsd
         http://odrl.net/1.1/ODRL-DD ../schemas/ODRL-DD-11.xsd">
   <o-ex:agreement>
      <o-ex:asset>
         <ds:keyInfo>
            <ds:keyValue>%KEY%</ds:keyValue>
         </ds:keyInfo>
         <o-ex:context>
            <o-dd:uid>%CID%</o-dd:uid>
            <o-dd:name>%PARAM_1%</o-dd:name>
         </o-ex:context>
      </o-ex:asset>
      <o-ex:permission>
         <o-dd:play>
            <o-ex:constraint>
               <o-dd:individual>%UID%</o-dd:individual>
               <o-dd:count>%PARAM_2%</o-dd:count>
               <o-dd:datetime>
                <o-dd:start>%SDATE%</o-dd:start>
                <o-dd:end>%EDATE%</o-dd:end>
               </o-dd:datetime>
            </o-ex:constraint>
         </o-dd:play>
      </o-ex:permission>
   </o-ex:agreement>
</o-ex:rights>
```

Listing 1 - Example ODRL License template specific for a business model

On the license template all the parameters that can be replaced are represented using a specific notation (%KEY%, %CID%, %UID%, %SDATE, %EDATE, %PARAM%). The license production process works in the following way (Figure 5): (1) each of the content suppliers defines their own ODRL license templates, specifying business rules and conditions for each of the templates. When an end-user obtains protected content from some content supplier, a license is produced (3) using the specific license template defined previously, the content unique identifier and the user identifier (2). Afterwards, the license can be downloaded (4) by the end user

– not directly by the end-user but by the Wallet.

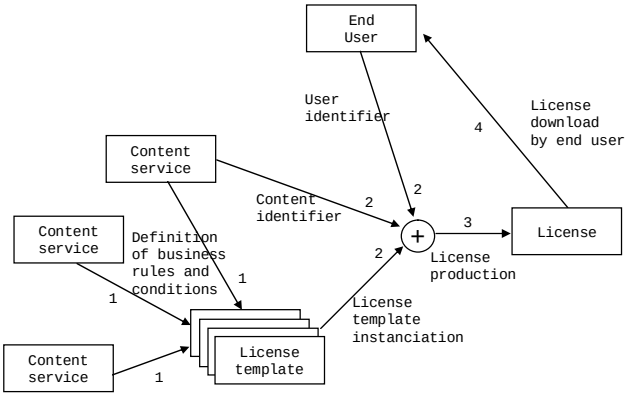


Figure 5 - Process to define a License

The described system is being used in several scenarios. These scenarios, described on the following sections, share the same License Server and the same digital Wallet at the client-side, but they have a different license template for each scenario: music download and streaming, video-surveillance and remote sensing JPEG2000 images.

For each of the scenarios, the most representative business model conditions have been identified and established on the License Server – in some cases more than one license template can be established for the same business scenario. In the case of the system presented here, the license templates are defined manually, but the production of web-based license template definition software is predicted that will allow content service providers (or content authors) to express their own rules on content usage in a very simple and natural way.

IV. USAGE SCENARIOS

All the scenarios that are presented in this paper use DRM and ODRL to control the access and conditions, of a given user or device to a particular content. Although these scenarios are quite different in nature, the used licenses share some commonalities (and at the same time some specific differences). In what concerns the commonalities among all the three proposed scenarios, they can be summarized in the following:

- Content identification (%CID%): each license contains the unique identifier which specifies that the license refers to a specific content [16][17], or content part;
- User identification (%UID%): all the licenses contain a way of specifying which user, group or domain is bound to the license;
- Expiry date (%SDATE%, %EDATE%): this parameter indicates the license validity period. This parameter supersedes the render content count (in case it exists), meaning that if the validity period expires before the counter reaches 0, the license is considered invalid;
- Content Encryption Key(s) (%KEY%): each of the licenses have one or multiple Content Encryption Key (CEK) that can be used by the appropriate end-user applications to access the protected content;
- License confidentiality and integrity: all the licenses (although not specified on the examples given on this paper) have the CEK ciphered in such a manner that can only be deciphered by the user's Wallet and all the licenses are digitally signed by the License Server to prevent their modification.

On the other hand, each of the proposed scenarios has specific conditions that are imposed on their license templates which were defined by the content providers. These conditions are specified and exemplified on the following sections.

A. *Music download and streaming*

This scenario, similar to others, represents a typical music portal, where an end-user can go and select some tracks of music to download/stream to listen [4]. OpenSDRM is used to control the access to the music and a specific license template was established for this particular scenario. This license template allows the specification of the following conditions:

- Play count: this parameter allows to setup how many times the content can be rendered by the end-user application;
- Operations: this parameter allows the definition of a set of possible operations that might be conducted over the content – in the case of the presented music business model the possible operations are: lend, save and play.

The following example (Listing 2) provides a sample license for the music download business model, with some of the generic and specific license parameters instantiated.

```
<?xml version="1.0" encoding="UTF-8" ?>
<o-ex:rights xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xmlns:o-dd="http://odrl.net/1.1/ODRL-DD"
         xmlns:ds="http://odrl.net/1.1/ODRL-DD"
         xsi:schemaLocation="http://odrl.net/1.1/ODRL-EX
../schemas/ODRL-EX-11.xsd
         http://odrl.net/1.1/ODRL-DD ../schemas/ODRL-DD-11.xsd">
   <o-ex:agreement>
      <o-ex:asset>
         <ds:keyInfo>
            <ds:keyValue>%KEY%</ds:keyValue>
         </ds:keyInfo>
         <o-ex:context>
            <o-dd:uid>%CID%</o-dd:uid>
            <o-dd:name>Call On Me</o-dd:name>
         </o-ex:context>
      </o-ex:asset>
      <o-ex:permission>
         <o-dd:lend/>
         <o-dd:play>
            <o-ex:constraint>
               <o-dd:individual>%UID%</o-dd:individual>
               <o-dd:count>%PARAM_1%</o-dd:count>
               <o-dd:datetime>
                <o-dd:start>%SDATE%</o-dd:start>
                <o-dd:end>%EDATE%</o-dd:end>
               </o-dd:datetime>
            </o-ex:constraint>
         </o-dd:play>
      </o-ex:permission>
   </o-ex:agreement>
</o-ex:rights>
```

Listing 2 - Example of ODRL license for the music download scenario

This scenario was developed during an IST RTD project called MOSES [22], in a specific trial which targeted the

electronic commerce of digital music. This trial exploited a service called Music-4You (Figure 6), which allowed the users to obtain music and acquire the respective licenses. It used a dynamic price adjustment mechanism that established the final price according to the usage conditions selected by the final user.

Figure 6 – The Music-4You web-site

B. Video-surveillance streaming and storage

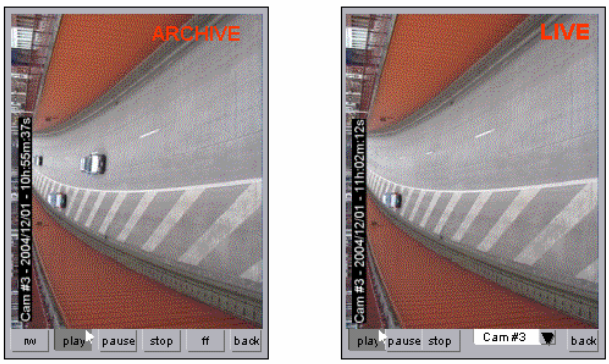
This scenario aims at the development of an integrated system for secure delivery of video surveillance data over a wireless network, while remaining scalable and robust to transmission errors. To achieve these goals, the content is encoded in Motion-JPEG2000 [21] and streamed with a specific RTP [27] protocol encapsulation to prevent the loss of packets containing the most essential data. Protection of the video data is performed at content level using the standardized JPSEC syntax [20], along with flexible encryption of quality layers or resolution levels. OpenSDRM is used to manage all authenticated peers on the WLAN (from end-users to cameras), as well as to manage the rights to access and display conditionally the video data. The OpenSDRM License Server produces licenses for this scenario based on the following parameters:

- Resolution level: the video-surveillance data maybe streamed with different quality resolution layer. The license defined in these scenarios allows the definition of different access levels concerning the resolution layer;
- Operations: this parameter allows the specification of the possible operations that can be conducted over the content by a given user or group of users: save, display or play.

The following example (Listing 3) provides a sample license for the video-surveillance streaming business model, with some of the generic and specific license parameters instantiated.

```
<?xml version="1.0" encoding="UTF-8" ?>
<o-ex:rights xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:o-dd="http://odrl.net/1.1/ODRL-DD"
        xmlns:ds="http://odrl.net/1.1/ODRL-DD"
        xsi:schemaLocation="http://odrl.net/1.1/ODRL-EX
../schemas/ODRL-EX-11.xsd
        http://odrl.net/1.1/ODRL-DD ../schemas/ODRL-DD-11.xsd">
   <o-ex:agreement>
      <o-ex:asset>
         <ds:keyInfo>
            <ds:keyValue>%KEY_1%</ds:keyValue>
         </ds:keyInfo>
         <o-ex:context>
            <o-dd:uid>%CID_1%</o-dd:uid>
         </o-ex:context>
      </o-ex:asset>
      <o-ex:permission>
         <o-dd:save/>
         <o-dd:display>
            <o-ex:constraint>
               <o-dd:datetime>
                <o-dd:start>%SDATE%</o-dd:start>
                <o-dd:end>%EDATE%</o-dd:end>
               </o-dd:datetime>
            </o-ex:constraint>
         </o-dd:display>
         <o-dd:play>
            <o-ex:constraint>
               <o-dd:datetime>
                <o-dd:start>%SDATE%</o-dd:start>
                <o-dd:end>%EDATE%</o-dd:end>
               </o-dd:datetime>
            </o-ex:constraint>
         </o-dd:play>
      </o-ex:permission>
   </o-ex:agreement>
   <o-ex:agreement>
      <o-ex:asset>
         <ds:keyInfo>
            <ds:keyValue>%KEY_2%</ds:keyValue>
         </ds:keyInfo>
         <o-ex:context>
            <o-dd:uid>%CID_2%</o-dd:uid>
         </o-ex:context>
      </o-ex:asset>
      <o-ex:permission>
         <o-dd:save/>
         <o-dd:display>
            <o-ex:constraint>
               <o-dd:group>%UID%</o-dd:group>
               <o-dd:datetime>
                <o-dd:start>%SDATE%</o-dd:start>
                <o-dd:end>%EDATE%</o-dd:end>
               </o-dd:datetime>
            </o-ex:constraint>
         </o-dd:display>
         <o-dd:play>
            <o-ex:constraint>
               <o-dd:group>%UID%</o-dd:group>
               <o-dd:datetime>
                <o-dd:start>%SDATE%</o-dd:start>
                <o-dd:end>%EDATE%</o-dd:end>
               </o-dd:datetime>
            </o-ex:constraint>
         </o-dd:play>
      </o-ex:permission>
   </o-ex:agreement>
</o-ex:rights>
```

Listing 3 - Example of ODRL license for the video-surveillance streaming



A B

Figure 7 – WCAM prototype application

This scenario is currently under development by the FP6 IST RTD project called WCAM [23]. The system prototype has already been presented at the end of the first year of the project (Figure 7), and will be continuously improved towards its testing during the next Annecy 2005 International Animated Festival [24], where a live trial will be conducted to

the system.

C. *Remote sensing of JPEG2000 images*

The third and final scenario that will be presented in this paper refers to a content business situation in which an end-user can access an Earth Observation (EO) portal on the WWW and order some visible EO products which are then converted to JPEG2000 images [18][19]. These JPEG2000 EO products are protected by the EO portal supplier and sent in an encrypted format (using the JPSEC format) to the end-user. OpenSDRM is used to protect the access to the multiple resolutions of the EO product and to control which operations can be conducted over the content. OpenSDRM produces licenses for the EO products based on a template that allows the specification of the following parameters:

- Resolution level: the JPEG2000 EO products have different resolutions (to a maximum number of six). Each of the resolutions is protected with a different key and the access to each level can be conditioned to a particular user or user group;
- Operations: this parameter allows the specification of which are the operations that can be conducted on the content. In this particular business model the save operation is the one that is possible to specify. This operation allows the end-user to recover the original EO product format.

The following example (Listing 4) provides a sample license for the remote sensing images business model, with some of the generic and specific license parameters instantiated.

```
<?xml version="1.0" encoding="UTF-8" ?>
<o-ex:rights xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xmlns:o-dd="http://odrl.net/1.1/ODRL-DD"
         xmlns:ds="http://odrl.net/1.1/ODRL-DD"
         xsi:schemaLocation="http://odrl.net/1.1/ODRL-EX
../schemas/ODRL-EX-11.xsd
         http://odrl.net/1.1/ODRL-DD ../schemas/ODRL-DD-11.xsd">
   <o-ex:agreement>
      <o-ex:asset>
         <ds:keyInfo>
            <ds:keyValue>%KEY_1%</ds:keyValue>
         </ds:keyInfo>
         <o-ex:context>
            <o-dd:uid>%CID_1%</o-dd:uid>
         </o-ex:context>
      </o-ex:asset>
      .
      .
      .
      <o-ex:asset>
         <ds:keyInfo>
            <ds:keyValue>%KEY_6%</ds:keyValue>
         </ds:keyInfo>
         <o-ex:context>
            <o-dd:uid>%CID_6%</o-dd:uid>
         </o-ex:context>
      </o-ex:asset>
      <o-ex:permission>
         <o-dd:display>
            <o-ex:constraint>
               <o-dd:individual>%UID%</o-dd:individual>
               <o-dd:datetime>
                <o-dd:start>%SDATE%</o-dd:start>
                <o-dd:end>%EDATE%</o-dd:end>
               </o-dd:datetime>
            </o-ex:constraint>
         </o-dd:display>
         <o-dd:display>
            <o-ex:constraint>
               <o-dd:individual>%UID%</o-dd:individual>
               <o-dd:datetime>
                <o-dd:start>%SDATE%</o-dd:start>
                <o-dd:end>%EDATE</o-dd:end>
               </o-dd:datetime>
            </o-ex:constraint>
         </o-dd:display>
      </o-ex:permission>
   </o-ex:agreement>
</o-ex:rights>
```

Listing 4 - Example of ODRL license for the remote sensing scenario

This scenario was developed during a European Space Agency (ESA) project, called HICOD2000 [25]. HICOD2000 implemented this scenario that allowed the service provider to protect the EO products and to define at the same time licenses which controlled the end-user access to such products. This system was implemented and was integrated within ESA EO products portal.

The users can browse EO products from the ESA portal, select the products and the corresponding resolution level, and perform its payment. The ESA portal connects the EO product service provider that produces the JPEG2000 version of the EO product and protects it.

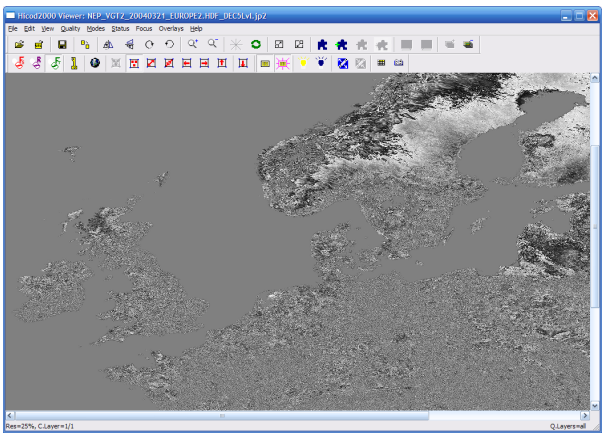


Figure 8 – The HICOD2000 Viewer

The EO service provider uses the OpenSDRM platform to specify the licenses of each of the EO products. When the user receives the EO product he can open it on a specific viewer (Figure 8) that enforces the license over the content (which is protected).

V. CONCLUSIONS

In this paper we have described a system that uses ODRL to express rights over protected content [14]. This system, referred to as OpenSDRM [3][4], uses a mechanism that enables interoperability at the client-side of the different protected content types and different content applications – the OpenSDRM digital Wallet. This mechanism enables DRM-supported applications to request, to the digital Wallet middleware, authorization to perform operations over the protected content. The required clearance of these operations, mediated by the Wallet, is expressed in ODRL-formatted licenses [14]. However, the system is REL-independent.

The system is based on the notion of license templates, which are defined taking into account the content business rules expressed by the content supplier. The presented system enables a multiplicity of different license conditions for different content suppliers, since, whenever a license is issued

to a given user, the license server instantiates a license template with the appropriate parameters.

The paper has also provided three different usage scenarios in which the system is being used, demonstrating its applicability and usefulness in mediating the access to digital music, remote sensing images and video-surveillance streams. These three different scenarios share the same License server with three different ODRL license templates. The License server, according to the content service, issues a specific license (instantiating the template) that can subsequently be downloaded after by the license system middleware, present at the end user client. This client-side middleware, the OpenSDRM digital Wallet, receives requests from the applications to be granted access to operations with the content.

REFERENCES

[1] Chiariglione, L., "Intellectual Property in the Multimedia Framework", Management of Digital Rights, Berlin (2000)

[2] Duhl J., Keroskian S., "Understanding DRM Systems", IDC White Paper, 2003

[3] Serrão C., "Open Secure Infrastructure to control User Access to multimedia content", WEDELMUSIC2004, September 2004, Barcelona

[4] Serrão C., Neves D., Kudumakis P., Barker T., Balestri M., "OpenSDRM - An Open and Secure Digital Rights Management Solution", IADIS 2003, Lisboa,

[5] Prunela A., "Windows Media Technologies: Using Windows Media Rights Manager to Protect and Distribute Digital Media", MSDN Magazine, December (2001), http://msdn.microsoft.com/msdnmag/issues/01/12/DRM/default.aspx

[6] Bormans J., Hill K., "MPEG-21 Overview v.5", ISO/IEC JTC1/SC29/WG11/N5231, 2002

[7] ISO/IEC 21000-3 Information technology -- Multimedia framework (MPEG-21) -- Part 3: Digital Item Identification

[8] Microsoft, "Architecture of Windows Media Rights Manager", Microsoft Corporation, http://www.microsoft.com/windows/windowsmedia/howto/articles/drmarchitecture.aspx, May 2004

[9] Lenzi, R. et al, "Apple iTunes Music Store", technical report, The Interactive-Music Network, June 2003

[10] "XML Signature Syntax and Processing", W3C Recommendation, February 2002, http://www.w3.org/TR/xmldsig-core/

[11] "XML Encryption Syntax and Processing", W3C Recommendation, December 2002, http://www.w3.org/TR/xmlenc-core/

[12] Microsoft, "Scenarios for Windows Media DRM", Microsoft Corporation, 2004, http://www.microsoft.com/windows/windowsmedia/drm/scenarios.aspx

[13] "SOAP Security Extensions: Digital Signature", W3C Note, February 2001, http://www.w3.org/TR/2001/NOTE-SOAP-dsig-20010206/

[14] "Open Digital Rights Language (ODRL) Version 1.1", W3C Note, September 2002, http://www.w3.org/TR/odrl/

[15] Iannella R., "Digital Rights Management (DRM) Architectures", D-Lib Magazine, Volume 7 Number 6 ISSN 1082-9873, June 2001 http://www.dlib.org/dlib/june01/iannella/06iannella.html

[16] Dalziel, J., "DOI in a DRM environment", White Paper, Copyright Agency Limited, 2004

[17] Rosenblatt, B. "Enterprise Content Integration with the Digital Object Identifier: A Business Case for Information Publishers", June 2002

[18] ISO/IEC 15444-1/ IUT-T T.800, JPEG2000 Image Coding System - Part 1: Core Coding System, 2000.

[19] D. Taubman and M. Marcellin, JPEG 2000: Image Compression Fundamentals, Standards and Practice, Kluwer Academic Publishers, 2002.

[20] JPSEC Final Committee Draft 1.0, ISO/IEC JTC1/SC29 WG1 N3480, November 2004

[21] Sadourny, Y., Conan, V., Serrão, C. Fonseca, P., "WCAM: secured video surveillance with Digital Rights Management", WCAM project, SPIE, 2004

[22] MOSES web-site, http://www.ist-moses.org

[23] WCAM web-site, http://www.ist-wcam.org

[24] Annecy 2005 International Animated Festival web-site, http://www.annecy.org

[25] HICOD2000 web-site, http://www.hicod2000.org

[26] Music-4You web-site, http://www.music-4you.com

[27] Schulzrinne H., Casner S., Frederick R., Jacobson V., „RTP: A Transport Protocol for Real-Time Applications", RFC1889, January 1996

Formalising ODRL Semantics using Web Ontologies

Roberto García, Rosa Gil, Isabel Gallego and Jaime Delgado

***Abstract*—In order to move Digital Rights Management to the Internet, a common rights expression language is needed. ODRL (Open Digital Rights Language) is one of the proposed solutions. It is based on a XML language and thus it just formalises the language syntax, while language semantics are specified informally. Actually, ODRL seems quite complete and generic enough to cope with such a complex domain. However, the problem is that it has such a rich structure that it is difficult to implement. In our opinion, it lacks formal semantics that would help ODRL applications development.**

As the application context is the Web, our approach to formalise ODRL semantics is based on semantic web ontologies. Firstly, ORDL has been moved to the Semantic Web space using XML Schema to OWL and XML to RDF tools. This provides some simple semantics. In order to refine them, the resulting ODRL ontologies have been connected to IPROnto, a result of previous research.

IPROnto, Intellectual Property Rights Ontology, models the IPR core concepts for creation, intellectual property rights and the basic kinds of actions that operate on intellectual property. It enables semantics-aware IPR applications that benefit from semantic queries, in contrast to the difficulties that emerge from the use of syntactic queries when the information space is as complicated as in the IPR field. Moreover, specialised reasoners can be used for license checking and retrieval. All these advantages have been propagated to ODRL thanks to this mapping.

***Index Terms*—Copyright protection, Digital Rights Management, Knowledge representation, Ontology**

I. Introduction

THE amount of digital content delivery in the Internet has made Web-scale Digital Rights Management (DRM) a key issue. Traditionally, DRM Systems (DRMS) have deal with this problem for bounded domains. However, when scaled to the Web, DRMSs are very difficult to develop and maintain. The solution is interoperability of DRMS, i.e. a common framework for understanding that defines a shared rights expression languages and its associated vocabulary.

This work was supported in part by Agent Web, Spanish administration TIC 2002-01336, and VISNET, European Commission IST FP6 Network of Excellence (http://www.visnet-noe.org).

Roberto García, Rosa Gil and Jaime Delgado are with the Technology Department, Universitat Pomeu Fabra, Barcelona, E-08003 Spain (e-mail: roberto.garcia, rosa.gil, jaime.delgado@upf.edu).

Isabel Gallego is with the Computer Architecture Department, Universitat Politècnica de Catalunya, Barcelona, E-08034 Spain (e-mail: isabel.gallego@ac.upc.edu).

ODRL (Open Digital Rights Language – http://odrl.net) [1] is one possible approach to that. It is a XML language defined by two XML Schemas. The first XML Schema defines the language syntax and a basic vocabulary. The second XML schema is called the Data Dictionary. It provides the complete vocabulary with textual definitions and a lightweight formalisation of the vocabulary terms semantics as an XML Schema.

ODRL seems quite complete and generic enough to cope with such a complex domain. However, the problem is that it has such a rich structure that it is difficult to implement. It is rich in the context of XML languages and the "traditional" XML tools like DOM or XPATH. There are too many attributes, elements and complexTypes, see Table 1, to deal with.

Table 1. Number of named XML Schema primitives in ODRL

	ODRL	
Schema	EX-11	DD-11
xsd:attribute	10	3
xsd:complexType	15	2
xsd:element	23	74
Total	127	

For instance, consider looking for all constraints in a right expression that apply to how we can access the licensed content. This would require so many XPATH queries as there are different ways to express constraints. ODRL defines 23 constraints: industry, interval, memory, network, printer, purpose, quality… This amounts to lots of source code, difficult to develop and maintain because it is very sensible to minor changes to the ODRL specification. Fortunately, there is a workaround hidden in the language definitions.

As we have said, there is the language syntax but also some semantics. The substitutionGroup relations among elements and the extension/restriction base ones among complexTypes encode generalisation hierarchies that carry some lightweight, taxonomy-like, semantics.

For instance, all constraints in ODRL are defined as XML elements substituting the o-ex:constraintElement. The difficulty is that although XML Schemas provide this information, it remains hidden when working with instance documents of this XML Schemas.

Moreover, there are more complex semantics encoded in the textual definitions of the Rights Data Dictionary. They are needed each time a programmer is developing an ODRL

application and thus they must be “manually” interpreted repeatedly.

Our idea is to make the ODRL semantics explicit in order to exploit ODRL hidden semantics and to attach more complex formalisations that facilitate ODRL applications implementation. This objective can be accomplished using ontologies and we have already tested it in the context of rights expression languages, concretely for the formalisation of the MPEG-21 Rights Data Dictionary semantics [2].

Ontologies are formalisations of a shared conceptualisation. They are formal so they provide the required semantics in a machine-readable form. They can be used to provide the required definitions of the rights expression language terms in a formal form. Thus, from the automatic processing point of view, a more complete vision of the application domain is available and more sophisticated processing can be carried out.

In the Web context, ontologies are promoted by the Semantic Web initiative [3] as a tool for Web-wide semantics-enabled processing. We have taken the Semantic Web approach because it is naturally prepared for the Internet domain and thus we use web ontologies [4].

The main Semantic Web languages are RDF for semantic metadata and OWL for web ontologies. They are introduced in section II. Their relation is analogous to the one between XML for metadata and XML Schema for metadata structuring, although in a semantic, and not only syntactic, information space.

We will use OWL as the tool to formalise ODRL semantics. This formalisation will be accomplished in two phases. First, the lightweight semantics encoded in the ODRL XML Schemas will be translated to OWL ontologies that make them explicit. This is detailed in section III.

Second, it is time for the data dictionary semantics informally written down as textual definitions. It is difficult to formalise them but even if the formalisation is incomplete, they will greatly facilitate ODRL applications development. A preliminary attempt in this direction is shown in section IV.

II. SEMANTIC WEB LANGUAGES OVERVIEW

The Semantic Web paradigm is an attempt to leverage the Web from a distributed information repository to a distributed knowledge one. The Semantic Web basic tools are the Resource Description Framework (RDF) [5] and RDF Schema [6]. A more advanced tool is the Web Ontology Language (OWL) [7].

RDF is used to associate metadata to resources in order to make information about them explicit. Resources are named using URIs, i.e. URLs or URNs. The RDF modelling primitive is the graph. It is composed by a set of arcs used to assert property values about resources and to relate resources between them. Arcs are also called triples in RDF terminology. Each graph arc is composed by a subject URI (the resource about which the statement is made), a property URI and a value (literal) or object URI (the resource to which the subject is related by the property). An RDF description is composed by a set of arcs describing some resources. The set of arcs constitutes a graph that can be navigated in order to retrieve the desired metadata.

As it has been seen until now, RDF provides a framework to model metadata. The basic primitive is the graph. This can be compared with the XML context, where the modelling tool is the tree. However, as an XML tree, an RDF graph is on its own basically unrestricted. Therefore, in order to capture the semantics of a particular domain, some primitives to build concrete “how things are connected” restrictions are necessary.

The tool that provides these restriction-building primitives is RDF Schema. It can be compared to XML Schema or DTDs, which provide building blocks to define restrictions about how XML elements and attributes are related. The primitives are some restricted URI names defined in the RDF and RDFS namespaces. RDFS provides Object Orientation-like primitives. With these primitives, class hierarchies can be defined. Resources are declared members of some of these classes and inherit their associated restrictions.

Moreover, there is a special kind of class: Property. It contains all the resources used to relate subject and object in triples, i.e. all the resources used to name the graph arcs. Property hierarchies can also be defined, and domain (origin) and range (destination) of the RDF graph arcs can be restricted to specific classes.

The Web Ontology Language (OWL) is a more advanced ontology-building toolkit. It provides more fine-grained primitives that allow additional restrictions. OWL is superset of RDF/S, i.e. in an OWL ontology all the primitives of RDF/S can be used.

III. MAKING ODRL XML SCHEMAS SEMANTICS EXPLICIT

As we have said, XML Schemas define the ODRL language syntax but also some simple semantics. The substitution group relations among elements and the extension/restriction base ones among complex types encode generalisation hierarchies.

There are many attempts to make XML metadata semantics explicit, usually they translate it to Semantic Web languages that facilitate the formalisation. Some of them just model the XML tree using the RDF primitives [8]. Others concentrate on modelling the knowledge implicit in XML languages definitions, i.e. DTDs or the XML Schemas, using web ontology languages [9], [10], [11]. Finally, there are attempts to encode XML semantics integrating RDF into XML documents [12], [13].

However, none of them facilitates an extensive transfer of XML metadata to the Semantic Web in a general and transparent way. Their main problem is that the XML Schema implicit semantics are not made explicit when XML metadata instantiating this schemas is mapped. Therefore, they do not take profit from the XML semantics and produce RDF metadata almost as semantics-blind as the original XML. Alternatively, they capture this semantics but they use

additional ad-hoc semantic constructs that produce less transparent metadata.

Therefore, we have chosen the ReDeFer methodology [14] that combines a XML Schema to web ontology mapping, called XSD2OWL, with a transparent mapping from XML to RDF, XML2RDF. The ontologies generated by XSD2OWL are used during the XML to RDF mapping in order to generate semantic metadata that makes XML Schema semantics explicit. Both steps are detailed next and then their application to ODRL is shown.

A. *XSD2OWL Mapping*

The XML Schema to OWL mapping is responsible for capturing the schema implicit semantics. This semantics are determined by the combination of XML Schema constructs. The XSD2OWL mapping is based on translating this constructs to the OWL ones that best capture their semantics. These translations are shown in Table 2.

The XSD2OWL mapping is quite transparent and captures a great part of XML Schema semantics. The same names used for XML constructs are used for OWL ones, although in the new namespace defined for the ontology. Therefore, it produces OWL ontologies that make explicit the semantics of the corresponding XML Schemas. The only caveats are the implicit order conveyed by xsd:sequence and the exclusivity of xsd:choice.

Table 2. XSD2OWL translations for the XML Schema constructs and shared semantics with OWL constructs

XML Schema	OWL	Shared informal semantics
element|attribute	owl:DatatypeProperty| ObjectProperty	Named relation between nodes or nodes and values
element @substitutionGroup	rdfs:subPropertyOf	Relation can appear in place of a more general one
element@type	rdfs:range	The relation range kind
complexType| group|attributeGroup	owl:Class	Relations and contextual restrictions package
complexType// element	owl:Restriction	Contextualised restriction of a relation
extension|restriction @base	rdfs:subClassOf	Package concretises the base package
@maxOccurs| @minOccurs	owl:maxCardinality| minCardinality	Restrict the number of occurrences of a relation
sequence|choice	owl:intersectionOf| unionOf	Combination of relations in a context

For the first problem, owl:intersectionOf does not retain its operands order. There is no clear solution that retains the great level of transparency that has been achieved. The use of RDF Lists might impose order but introduces ad-hoc constructs not present in the original metadata. Moreover, as it has been demonstrated in practise, the elements ordering does not contribute much from a semantic point of view. For the second problem, owl:unionOf is an inclusive union, the solution is to use the disjointness OWL construct, owl:disjointWith, between all union operands in order to make it exclusive.

B. *XML2RDF Mapping*

Once all the metadata XML Schemas are available as OWL ontologies, it is time to map the XML metadata that instantiates them. The intention is to produce RDF metadata as transparently as possible. Therefore, a structure-mapping approach has been selected [15]. It is also possible to take a model-mapping approach [16]. XML model-mapping is based on representing the XML information set using semantic tools. This approach is better when XML metadata is semantically exploited for concrete purposes. However, when the objective is semantic metadata that can be easily integrated, it is better to take a more transparent approach.

Transparency is achieved in structure-mapping models because they only try to represent the XML metadata structure, i.e. a tree, using RDF. The RDF model is based on the graph so it is easy to model a tree using it. Moreover, we do not need to worry about the semantics loose produced by structure-mapping. We have formalised the underlying semantics into the corresponding ontologies and we will attach them to RDF metadata using the instantiation relation rdf:type.

The structure-mapping is based on translating XML metadata instances to RDF ones that instantiate the corresponding construct in OWL. The more basic translation is between relation instances, from xsd:elements and xsd:attributes to rdf:Properties. Concretely, owl:ObjectProperties for node to node relations and owl:DatatypeProperties for node to values relations. Values are kept during the translation as simple types and RDF blank nodes are introduced in the RDF model in order to serve as source and destination for properties. They will remain blank until they are enriched with semantic information. For the moment, the current state of the mapping is shown in Fig. 1.

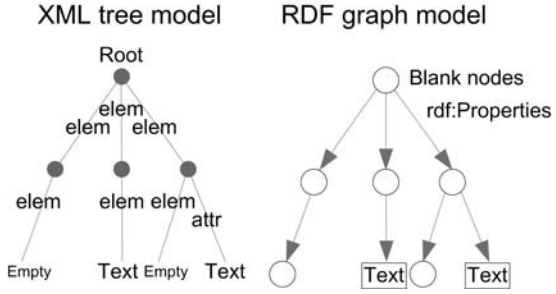


Fig. 1. XML tree and resulting RDF graph models

The current RDF graph model contains all that we can obtain from the XML tree. It is already semantically enriched thanks to the rdf:type relation that connects each RDF property to the owl:ObjectProperty or owl:DatatypeProperty it instantiates. It can be enriched further if the blank nodes are related to the owl:Class that defines the package of properties and associated restrictions they contain, i.e. the XML Schema complexTypes. This semantic decoration of the graph is formalised using rdf:type relations from blank nodes to the corresponding OWL classes.

At this point, we have obtained a semantics-enabled representation of the input metadata. The instantiation

relations can now be used to apply OWL semantics to metadata.

C. Application to ODRL XML Schemas

First of all, the XSD2OWL mapping has been applied to the ODRL XML Schemas. ODRL schemas define a quite flat set of hierarchies for complexTypes and elements. They are translated to OWL classes and properties hierarchies as shown in Fig. 2 and Fig. 3 respectively.

Once in OWL form, the previously hidden semantics can be exploited by OWL-aware tools that facilitate implementing ODRL applications.

Applications usually operate over ODRL instances, i.e. XML documents instantiating the XML Schemas. Therefore, in order to take profit from the just formalised semantic, it is necessary to map the XML instances to the semantic enriched form, i.e. to RDF metadata that instantiates the OWL ontologies just created.

The XML2RDF mapping resolves this. It receives the XML metadata for ODRL rights expressions and produces the RDF graph that models the corresponding XML tree. As it has been shown, the RDF graph is enriched with the XML Schema hidden semantics. Now, Semantic Web tools can easily put the ODRL XML Schemas semantics into practice.

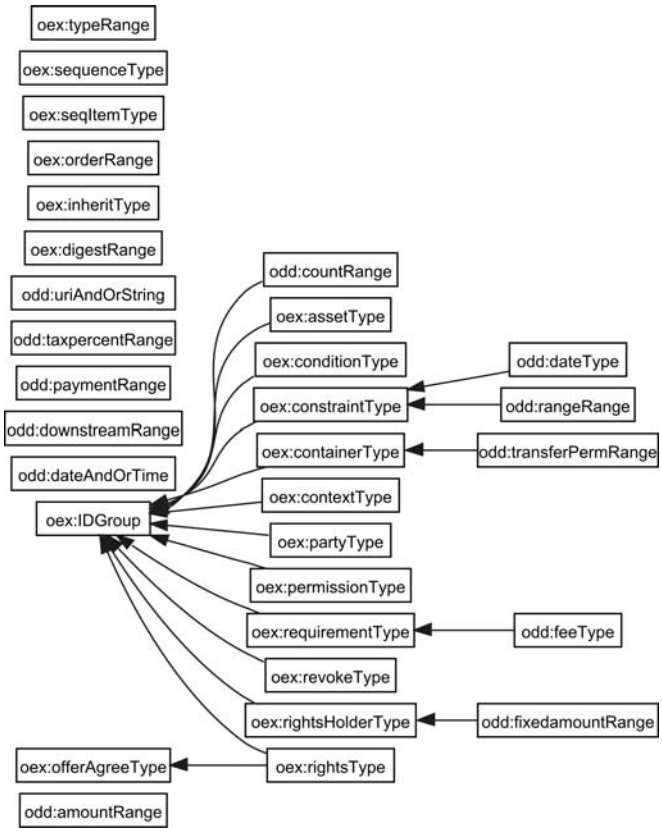


Fig. 2. ODRL XML complexTypes formalised as OWL classes hierarchies. The "Range" suffixed classes correspond to implicit complexTypes

For instance, we will retake the introduction problem about a query for retrieving the constraints affecting a ODRL rights expression. When we are working with the XML version, we need 23 XPath queries in order to retrieve all possible kinds of constraints. However, with the RDF version connected to the ODRL ontologies, a semantic query for o-ex:constraintElement will be automatically propagated in order to retrieve all the particular constraints defined as substitutionGroups.

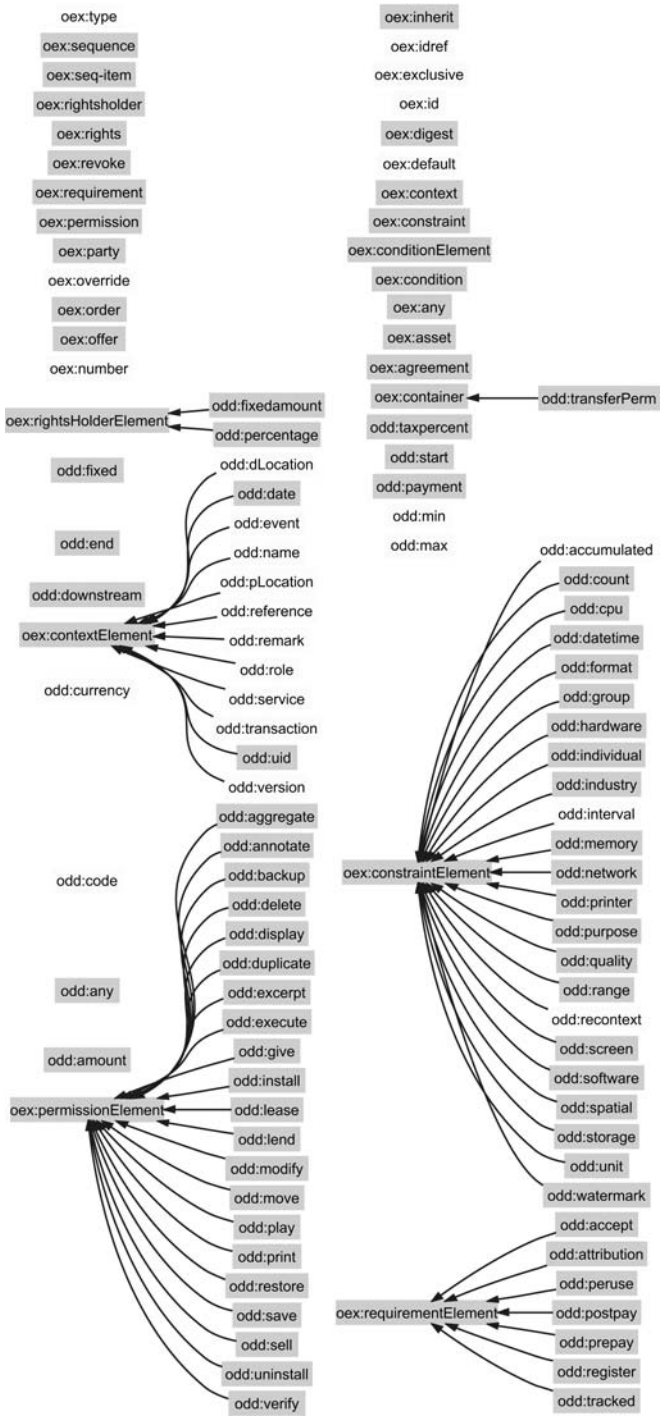


Fig. 3. ODRL XML elements and attributes formalised as OWL properties hierarchies. Grey properties correspond to object properties and white ones to datatype properties

D. Mapping results

As a result of the first step of ODRL semantics formalisation shown in this section, we have a methodology and some tools that allow us translating XML ODRL rights expressions into RDF-OWL.

The ODRL OWL ontologies formalise the XML Schema implicit semantics so they are available for Semantic Web tools in order to facilitate ODRL applications implementation. The ODRL Ontologies and metadata examples related to this section are available at [17].

Moreover, the ontologies will serve as the anchor point where more detailed semantics will be attached during the second step of ODRL semantics formalisation. This process is detailed in the next section.

IV. ODRL FORMALISATION USING AN IPR ONTOLOGY

The first step of ODRL semantics formalisation provides the lightweight semantics implicit in ODRL XML Schemas. Moreover, it provides the anchor points where we are going to attach the more detailed semantics formalised from the textual definitions of the Data Dictionary. The detailed semantics are written down as text so, in order to automatically extract them we would need natural language processing (NLP) methods. However, NLP techniques are not advanced enough to fully extract the intended semantics from the short descriptions of the Data Dictionary.

We use a different approach. An accurate reading of the definitions together with the whole ODRL specification will be done, i.e. automatic means are not used. This reading is intended for interpreting ODRL semantics in the framework of an Intellectual Property Rights Ontology, IPROnto [18, 19].

IPROnto is also a OWL web ontology that provides a general semantic framework for the Intellectual Property Rights (IPR) domain. IPROnto is presented in section IV.A. IPROnto guides the formalisation of ODRL semantics. The ODRL ontologies are connected to IPROnto following the interpretation of the ODRL specification. These mappings are detailed in section IV.B and IV.C. Finally, the benefits of the IPROnto-assisted formalisation of ODRL semantics are presented in section IV.D.

A. IPROnto

IPROnto is an ontology that tries to formalise the IPR domain from a general and purpose independent point of view. The ontology covers more than just the end user part of the intellectual property value chain. IPROnto models the full value chain and thus it must consider also the intellectual property rights part and not just the usage one. Moreover, it is not restricted to digital media. Therefore, it considers the general creation concept in detail as it is shown next.

IPROnto is firstly based on Intellectual Property literature and regulations, mainly from the World Intellectual Property Organisation (WIPO, http://www.wipo.org). The different IP aspects of IPROnto are detailed in the next subsections.

1) Creation Model: the core concepts of IPROnto are those that formalise the notion of creation. As we can see in Fig. 4, there are three points of view of a creation: the abstraction, manifestation and expression perspectives.

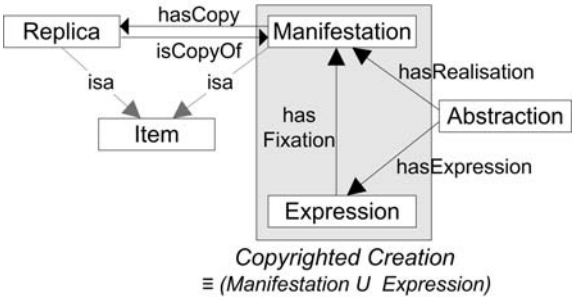


Fig. 4. Creation Model

For instance, if we take the creation “Les Misérables”, we can observe it from these three perspectives taking different forms. From the manifestation view, we can see a script, a book, etc. Its film projection would be seen from the expression perspective. All have in common the original Victor Hugo’s idea visible from the abstraction perspective. The ideas cannot be copyrighted so they lay outside the copyrighted creation concept. Abstraction, on the other hand, is what we grasp as common in different manifestations, expressions or replicas and what allows us saying that they are the same creation.

2) *Rights Model*: from the legal point of view, WIPO recommendations have been followed and the intellectual property rights they define are present in IPROnto. Table 3 shows the included rights hierarchy starting from Copyright. There are also other intellectual property rights that are not shown, e.g. sui-generis rights, neighbor rights, etc. although they are unimportant in this context.

Table 3. Copyright hierarchy

Copyright
MoralRight
DisseminationRight
PaternityRight
RespectRight
WithdrawalRight
ExploitationRight
TransformationRight
AdaptationRight
TranslationRight
SubtitlingRight
CommunicationRight
BroadcastRight
PublicPerformanceRight
DistributionRight
RentalRight
ReproductionRight
FixationRight

The more important rights in the Digital Rights Management context are Exploitation Rights as they are related to productive and commercial aspects of intellectual property. Each of these rights defines a set of actions that can be done or not on a creation depending on the rights situation:

- *Transformation Right*: grants actions of type transform that produce a new creation, like adapt, translate, subtitle, etc.
- *Communication Right*: grants actions of type communicate, like broadcast, perform, make available (e.g. on the Internet), etc.
- *Distribution Right*: grants actions of type distribute, like sell, rent, etc. This right, and consequently the kind of actions it includes, only affects manifestations of a creation (e.g. compact disk, DVD, cassette, etc.).
- *Reproduction Right*: grants actions of type reproduce, like copy, fix (an expression into a manifestation, e.g. an opera into a CD), etc.

Moral rights are always hold by the creator and cannot be commercially exploited. Moreover, they are only fully considered in Continental-like IPR systems, i.e. legal system like those in the European Union. On the other hand, legal systems of the Anglo-Saxon kind do not consider them. Therefore, as they do not have commercial interest, moral rights are modelled but not detailed in IPROnto for the moment.

We can also identify two more kinds of actions that are related to intellectual property, although the mentioned rights do not cover them:

- *Transfer*: these are actions to move rights between rights holders and are related to the exploitation aspect of intellectual property rights, only exploitation rights can be transferred. End users do not hold rights so there are no transfers to them. There are also commercial actions, which are related to transfer actions. Commercial actions are offer, agree, counteroffer, post-agree, etc.
- *Use*: end users do not hold exploitation rights. They just consume creations, i.e. they use them. Uses are not covered by copyright. However, this does not mean that end users can do whatever they want, they should not realise actions that require copyright. Moreover, they might be subject to special conditions under which they have acquired the permission to use a creation (e.g. a film that can only be viewed a fixed number of times and thus is cheaper than a DVD reproduction).

The previous actions are associated to the different roles that take part in the creation' life cycle. Or, from the commercial point of view, it can be seen as the creation's value chain. Legal persons play these roles. Actions are shown as arrows in Fig. 5. The ovals represent the different roles; those at the source of the arrows perform the actions. The arrow destinations show the role that receives the responsibility over the creation once the action has been performed.

First of all, the creator acts and a new creation is produced. Automatically, there is a holder that gets rights on the creation. The ovals represent roles that might be played by the same person. Therefore, the rights holder can be the same person that acted as creator.

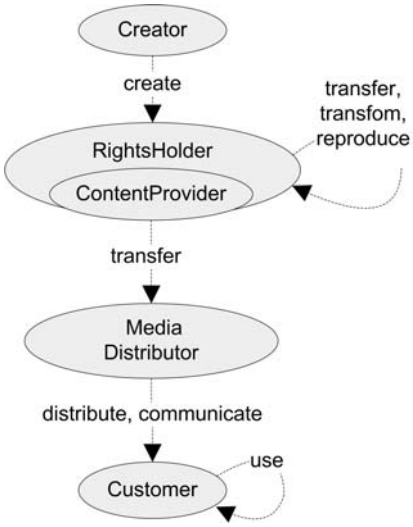


Fig. 5. Creation life cycle through the hands of the different roles involved and the actions they perform to move the creation forward

Then, the rights holder can transfer all or a portion of the rights to a content provider. Content providers are specialised in transforming raw creations in order to facilitate their commercialisation. Moreover, if the creation is commercialised physically, they are responsible for reproducing the creation in order to produce the replicas for consumption.

Next, it is time to make the creation available to end-users. Media distributors are responsible for this part. The get a transfer of the rights they need for the distribute and communicate actions, which are the actions that make creations available for end users.

Finally, at the end of the life cycle or value chain, the costumer uses the creation in order to consume it.

3) *IPROnto in "action"*: as it has been shown, IPROnto takes IP rights into account but it has actions as its central building block, where actions are those covered by exploitation rights but also usage and transfer ones. With them, we try to cover all the events in the value chain.

Actions are not isolated entities, they are related to a bunch of entities that take part or are affected by the action. Moreover, there are space-time coordinates that situate the action. One thing that all actions have in common is that they are verbs. Therefore, in order to facilitate their modelling, we have incorporated into IPROnto ideas from the linguistics field related to the classification of verbs and their relation to other linguistic components.

These relations are called thematic roles or case roles [20] and are classified into initiator, resource, goal and essence. In Table 4 we show the case roles we have considered in IPROnto and also the kinds of verbs they are related to. These kinds of verbs define verbs facets, not disjoint classes of

verbs, and concretise the general thematic roles as shown in each row. Therefore, the same verb can present one or more of these facets. For instance, the play verb can show the action, temporal and spatial facets in a particular sentence.

Table 4. General thematic roles (top row) and their concretisations corresponding to their relation to different verb facets (left column)

	initiator	resource	goal	essence
Action	agent, effector	instrument	result, recipient	patient, theme
Process	agent, origin	matter	result, recipient	patient, theme
Transfer	agent, origin	instrument, medium	experiencer, recipient	theme
Spatial	origin	path	destination	location
Temporal	start	duration	completion	pointInTime
Ambient	reason	manner	aim, consequence	condition

Fig. 6 shows an example of action modelling using thematic roles to relate the verb to its participants and context. In this case it is a reproduction of a master copy to produce CDs. It is done using a computer and is completed in 2000.

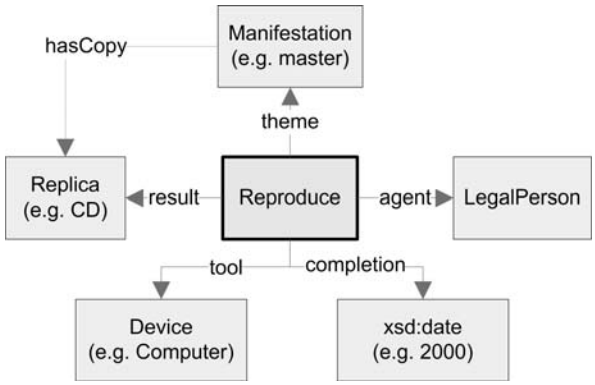


Fig. 6. Action modelling example using thematic roles

To conclude, IPROnto is enriched with general concepts for time, space, tools, part hood, etc. They are taken from upper level ontologies, which define general concepts. We need also specific concepts, e.g. digital media concepts, which are taken from domain ontologies. For instance, we have considered some upper ontologies and domain ontologies:

- Upper ontologies: IEEE SUMO [21], DOLCE [22] and LRI-Core [23]. They define general concepts; in the latter case with a clear legal bias. The other ones are general but include some legal aspects too.
- Domain ontologies: MPEG-7 ontology and TVAnytime ontologies. They are generated automatically from XML Schemas like ORDL ontologies.

B. Preparing ODRL Ontologies to IPROnto mappings

First of all, in order to facilitate mappings, some changes are introduced in the ODRL ontologies that were automatically generated from the ODRL XML Schemas. As it is shown in Fig. 2 and Fig. 3, elements are more richly structured than complexTypes. As a consequence, the OWL properties hierarchy is more complex than the OWL classes one.

The common situation for ontologies is the reverse one. Classes use to have richer hierarchical structure than classes and this is the case for IPROnto. Therefore, in order to facilitate mappings, the ODRL classes' hierarchy is enriched. We do not introduce any supplementary knowledge. The objective is simply to replicate the properties hierarchy structure in the classes' hierarchy.

The current lack of structure is because ODRL does not define more specific complexTypes for requirementType, permissionType and constraintType, since they are not needed while working with XML. On the other hand, the corresponding elements (requieremenElement, permissionElement and constraintElement) have more specific elements, which appear as their subproperties in the OWL ontology, i.e. play, software, prepay, etc.

Therefore, in order to replicate structure, we introduce a new class for each one of these properties and define the class as a subclass of the corresponding existing class. For instance, the PlayType class is introduced, corresponding to the play property, and it is defined as subclass of permissionType. The same is done for all the subproperties of requieremenElement, permissionElement and constraintElement.

The same applies for offer and agree, both related to the offerAgreeType complexType. The corresponding offerType and agreeType are introduced.

As the last preparatory step, we have also reintroduced in the ODRL ontologies all the abstract elements defined in the ODRL specification but not present in the XML Schemas. Consequenly, as detailed previously, we have also introduced the corresponding classes in order to replicate the new properties in the classes' hierarchy. They are use, reuse, transfer and asset management as permissionElement subproperties; interaction, fee and usage as requirementElement subproperties; user, device, bounds, aspect, target, temporal and rights as constraintElement subproperties.

C. Planning ODRL Ontologies to IPROnto mappings

Thanks to the previous preparatory step, we have new versions of ODRL ontologies that are easier to relate to IPROnto. We are currently planning the needed mappings in order to effectively produce the integration. It is work in progress so we are going to depict here the principles and techniques we are using. Moreover, we give some mapping examples.

The integration is performed using two techniques. First, for simple cases, it is possible to connect directly ontologies using OWL primitives for concept inclusion and equivalence (e.g. subClassOf, subPropertyOf, equivalentClass, equivalentProperty, sameIndividualAs, etc.).

These are some simple mapping examples (o-ex prefix refers to concepts generated directly from ODRL-EX, o-dd for ODRL-DD, o-ont for the extensions generated during the

previous preparatory step and ipro for concepts in IPROnto):

- o-ex:permissionType –subClassOf→ ipro:Verb
- oddo:usageType –subClassOf→ ipro:Use
- oddo:offerType –subClassOf→ ipro:Offer
- oddo:transferType –subClassOf→ ipro:Transfer
- o-dd:individual –subPropertyOf→ ipro:agent
- o-ex:asset –subPropertyOf→ ipro:essence
- o-dd:uid –equivalentProperty→ rdf:ID
- o-dd:name –equivalentProperty→ rdf:label
- etc.

However, the previous technique is only possible when we are mapping one concept from an ontology to one concept in the other ontology. When the conditions for the mapping are more complex, we are using semantic rules [24]. Rules are particularly useful when the mapping must cope with a difference in the manner the concepts are structured in the mapped ontologies.

For instace, the ODRL context element is not used in IPROnto. Web ontologies use the RDF identifier (rdf:ID) instead of the ORDL one (o-dd:uid) and RDF identifiers are directly attached to the concept they identify. In ODRL words this means that the identifier is a direct attribute of the asset. The same applies to the rest of the context model elements.

Therefore, the context element must be removed when mapping an ODRL instance to IPROnto. However, it is easier to convert the context of a contextualised type because it has all this information directly attached, while the contextualised type is empty. For instance, a contextualised description of an offer asset, see Fig. 7, is transformed using the previous simple mappings in conjunction with the mapping rule (1) to the IPROnto-aware description shown in Fig. 8.

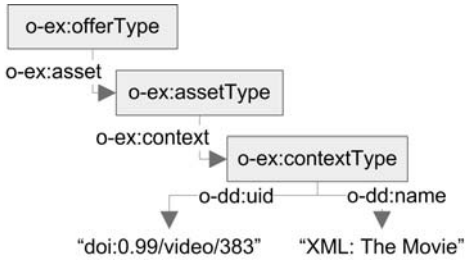


Fig. 7. ODRL example in RDF graph form

$$o\text{-}ex:asset(?x,?y) \wedge o\text{-}ex:assetType(?y) \wedge o\text{-}ex:context(?y,?z) \Rightarrow ipro:Creation(?z) \wedge o\text{-}ex:asset(?x,?z) \quad (1)$$

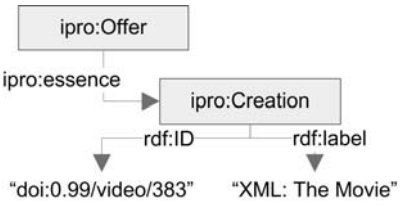


Fig. 8. IPROnto-aware graph resulting from mapping Fig. 7

D. IPROnto-ODRL benefits

The direct benefit of the ODRL to IPROnto mappings is that a substantial part of ODRL semantics are formalised. This might reduce ambiguities, or at least highlight possible ambiguous points. Moreover, there are new application development facilities. In addition to the semantic queries benefits shown before, other semantics-enabled tools can be used. One of the most promising tools is Description Logics (DL) [25]. OWL is based on DL so it can be directly fed into DL classifiers. Classifiers are specialised logic reasoners that guarantee computable results. DL classifiers are used with IPROnto in order to automatically check IP uses against the use patterns specified in IP agreements or offers. This facilitates checking if a particular use is allowed in the context of a set of licenses or finding an offer that enables it, once an agreement is reached.

DL classifiers can be directly reused so there is no need to develop ad-hoc applications to perform this function. Moreover, as they are completely OWL semantics aware, the IPROnto to ODRL ontologies mappings enables their use in order to check uses against ODRL licenses, even if they are in XML form. XML ODRL licenses can be mapped to RDF using XML2RDF and then, through mappings, get connected to the IPROnto semantic framework.

The use of DL classifiers for digital rights management, once mapped to IPROnto, can be exemplified with the following scenario:

1) The initial situation is: "USER1 is trying to access a given video stream from a streaming server at 9:30:10 UTC on 2005-04-10". The streaming server implements digital rights management. It inquires the license manager if the current usage is permitted. In order to do that, the streamer models this usage using IPROnto, see Fig. 9, and sends it to the license manager, e.g. as a RDF/XML serialisation.

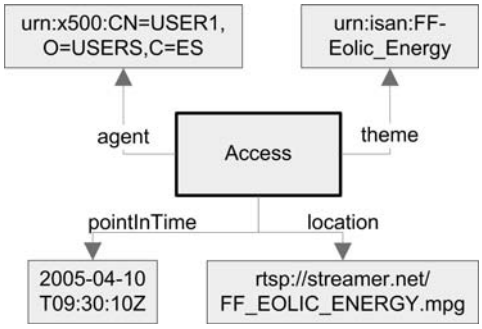


Fig. 9. Usage instance modelled by the streaming server

2) The license manager contains licenses modelled using IPROnto, including the one shown in Fig. 10. This license defines a usage pattern for a creation located at the streaming server that can be performed by a class of agents for a given period of time starting on a given date. Moreover, the license manager has additional metadata stating that USER1 is an instance of the "O=USERS,C=ES" class, which models a group of users.
3) The license manager checks if there is any license that grants a usage pattern that subsumes the usage instance. This can be performed easily and efficiently using a DL

classifier. However, there are some problems that should we resolved before. First, the usage patterns have a condition property that should be ignored during subsumption computation. Second, the usage patterns define time intervals using a start time and duration, while the usage instance defines a time point. In order to check if the time point is included in the time interval, we must use a DL classifier capable of dealing with custom datatypes reasoning [26]. Then, the time interval is translated to a real interval (2) and the time point to a real (3).

$$\text{pointInTime.}\geq[20050401]\ \text{real} \cap \leq[20060401]\ \text{real} \quad (2)$$

$$\text{pointInTime.}=[20050410.093010]\ \text{real} \quad (3)$$

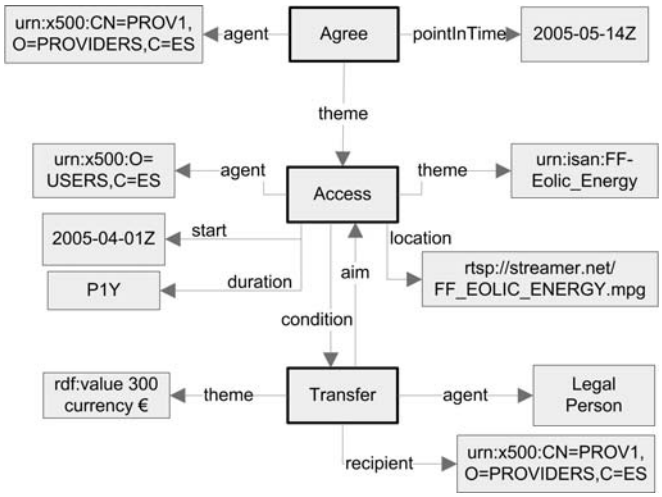


Fig. 10. Use license model defining permitted usage pattern and condition

4) After applying the previous adaptations, subsumption is computed. The usage might be classified in one or more usage patterns. In this case, we test if the usage pattern is the theme of an Agree event. This is equivalent to the agreement authorising this use. Finally, if the usage conditions are satisfied, the license manager tells the streaming server that the use is authorised. Otherwise, it is forbidden.

This is a simple scenario for illustrative purposes. It could be extended in many ways. For instance, if the usage pattern is the theme of an offer, another possibility is to recommend the user the possibility to negotiate it in order to arrive to a new agreement. From this point, this IPR reasoning framework can be connected to negotiation architectures previously developed in our research group [27, 28] in order to achieve assisted negotiation of digital goods.

V. Conclusion

As it has been shown, the Semantic Web approach to ODRL semantics formalisation has started to give its fruits. Even the first step of semantics formalisation, during which the implicit semantics of ODRL XML Schemas have been formalised, has proved very useful simply by making semantic queries possible.

The second step, during which more complex semantics are being defined, is showing promising results and it can greatly enlarge semantic benefits for ODRL applications implementation.

To conclude, it is important to remark that all this work has been done for the current version of ODRL, version 1.1. This version was intended for XML representation and this has made the connection of ODRL ontologies to IPROnto harder. For future versions of ODRL, it might be interesting to consider this possibility, which might enable a more complete formalisation using web ontologies.

References

[1] Iannella, R.: "Open Digital Rights Language (ODRL), Version 1.1". World Wide Web Consortium, W3C Note, 2002. Available: http://www.w3.org/TR/odrl

[2] Delgado, J.; Gallego, I. & García, R.: "Use of Semantic Tools for a Digital Rights Dictionary". In K. Bauknecht, K.; Bichler, M. & Pröll, B. (ed.) "E-Commerce and Web Technologies: 5th International Conference". Springer-Verlag, LNCS Vol. 3182, pp. 338-347, 2004

[3] Berners-Lee, T.; Hendler J. and Lassila O.: "The Semantic Web". Scientific American, May 2001. Available: http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21

[4] Hendler, J. "Agents on the Semantic Web". IEEE Intelligent Systems, Vol. 16, No. 2, March-April 2001. Available: http://www.ai.mit.edu/people/jimmylin/papers/Hendler01.pdf

[5] Lassila, O. and Swick, R.R. (eds.): "Resource Description Framework (RDF), Model and Syntax Specification". W3C Recommendation, 22 February 1999. Available: http://www.w3.org/TR/REC-rdf-syntax

[6] Brickley, D. and Guha, R.V. (eds.): "RDF Vocabulary Description Language 1.0: RDF Schema". W3C Working Draft, RDF Core Working Group, 2002. Available: http://www.w3.org/TR/rdf-schema

[7] Dean, M. and Schreiber, G. (eds.): "OWL Web Ontology Language Reference". W3C Candidate Recommendation, Web Ontology Working Group, 2003. Available: http://www.w3.org/TR/owl-ref

[8] Klein, M.C.A.: "Interpreting XML Documents via an RDF Schema Ontology". "Proceedings of the 13th International Workshop on Database and Expert Systems Applications (DEXA 2002)". pp. 889-894, 2002

[9] Amann, B.; Beeri, C.; Fundulaki, I. & Scholl, M.: "Ontology-Based Integration of XML Web Resources". "Proceedings of the 1st International Semantic Web Conference (ISWC 2002)". pp. 117-131, 2002

[10] Cruz, I.; Xiao, H. & Hsu, F.: "An Ontology-based Framework for XML Semantic Integration". "Eighth International Database Engineering and Applications Symposium". *Coimbra, Portugal*, 2004

[11] Halevy, A.Y.; Ives, Z.G.; Mork, P. & Tatarinov, I.: "Piazza: Data Management Infrastructure for Semantic Web Applications". "12th International World Wide Web Conference". 2003

[12] Lakshmanan, L. & Sadri, F.: "Interoperability on XML Data". "Proceedings of the 2nd International Semantic Web Conference (ICSW 03)". 2003

[13] Patel-Schneider, P.F. & Simeon, J.: "The Yin/Yang web: XML syntax and RDF semantics". "Proceedings of the 11th International World Wide Web Conference (WWW2002)". pp. 443-453, 2002

[14] ReDeFer, http://rhizomik.upf.edu/redefer

[15] Klein, M.C.A.: "Interpreting XML Documents via an RDF Schema Ontology". "Proceedings of the 13th International Workshop on Database and Expert Systems Applications (DEXA 2002)". pp. 889-894, 2002

[16] Tous, R. & Delgado, J.: "Using OWL for Querying an XML/RDF Syntax". Poster at WWW'05

[17] ODRL Ontologies, http://dmag.upf.edu/ontologies/odrlontos

[18] Delgado, J.; Gallego, I.; Llorente, S. & García, R.: "Regulatory Ontologies: An Intellectual Property Rights approach". In Meersman, R. & Tari, Z. (ed.): "On The Move to Meaningful Internet Systems 2003:

OTM 2003 Workshops". *Springer-Verlag*, Vol. 2889, pp. 621-634, 2003. ISBN: 3-540-20494-6

[19] Delgado, J.; Gallego, I.; Llorente, S. & García, R.: "IPROnto: An Ontology for Digital Rights Management". In Bourcier, D. (ed.): "Legal Knowledge and Information Systems". *IOS Press, Amsterdam*, Frontiers in Artificial Intelligence and Applications Vol. 106, 2003

[20] Sowa, J.F.: "Knowledge Representation. Logical, philosophical and computational foundations". *Brooks Cole Publishing Co.*, 2000

[21] Pease, A.; Niles, I. & Li, J.: "The Suggested Upper Merged Ontology: A Large Ontology for the Semantic Web and its Applications". In Pease, A. (ed.): "Ontologies and the Semantic Web, Papers from the AAAI Workshop". *AAAI Press*, 2002

[22] Gangemi, A.; Guarino, N.; Masolo, C.; Oltramari, A. & Schneider, L.: "Sweetening Ontologies with DOLCE". In Gómez-Pérez, A. & Benjamins, V. (ed.): "Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, 13th International Conference, EKAW 2002". *Springer-Verlag*, pp. 166-181, 2002

[23] Breuker, J.: "Constructing a legal core ontology: LRI-Core". In Proceeding of "Workshop on Ontologies and their Applications". *Sao Luis, Maranhao, Brazil*, 2004

[24] Horrocks, I.; Patel-Schneider, P.F.; Boley, H.; Tabet, S.; Grosof, B. & Dean, M.: "SWRL: A Semantic Web Rule Language Combining OWL and RuleML". *DARPA DAML Program*, Draft, No. 0.7, 2004. Available: http://www.daml.org/rules/proposal

[25] Pan, J.Z.: "Description Logics: "Reasoning Support for the Semantic Web". School of Computer Science, The University of Manchester, Oxford Rd, Manchester M13 9PL, UK, 2004

[26] Pan, J..Z. & Horrocks, I.: "OWL-Eu: Adding Customised Datatypes into OWL". In "Proc. of Second European Semantic Web Conference (ESWC 2005)". *Springer-Verlag*, 2005

[27] Delgado, J.; Gallego, I.; García, R. & Gil, R.: "An architecture for negotiation with mobile agents". In A. Karmouch; T. Magedanz & J. Delgado (ed.) "Mobile Agents for Telecommunication Applications: 4th International Workshop". *Springer-Verlag*, pp. 21-31, 2002

[28] Gil, R.; García, R. & Delgado, J.: "Delivery context negotiated by mobile agents using CC/PP". In "Mobile Agents for Telecom Applications, (MATA'03)". *Springer-Verlag*, pp. 99-110, 2003

Extending ODRL to Enable Bi-Directional Communication

Alapan Arnab and Andrew Hutchison
Data Network Architectures Group
Department of Computer Science
University of Cape Town
Rondebosch, 7701
South Africa
{aarnab, hutch}@cs.uct.ac.za

***Abstract*— Current rights expression languages (RELs) only allow for rights holders to dictate terms to the end users. This limits their use as a means for negotiating electronic contracts and end users are not able to request changes in their rights contracts. In this paper we propose extensions to ODRL that allow end users to request changes and for the rights holder to grant or deny these changes. These extensions allow the end user to request changes to their current rights, and for the rights holder to grant or refuse the request. We also provide two examples to demonstrate possible uses of our extensions. The extensions we discuss can also be implemented in other RELs like XrML.**

I. Introduction

Rights Expression Languages (RELs), like Open Digital Rights Language (ODRL) and eXtensible rights Markup Language (XrML), form an integral part of a DRM system because they allow the rights holders to express the terms and conditions which need to be upheld by DRM systems. Most RELs have an extensive vocabulary, supporting syntactic rules that allow them to express a variety of different terms and conditions. Thus RELs allow for greater flexibility in the expression of rights from the view of the rights holders.

However, RELs have also been criticised for giving rights holders too much control, and thus the flexibility offered by RELs empower only the rights holders and not the end users. This stems from the access control models used by most RELs – only rights expressed in the usage license are granted to the user, and thus rights not mentioned are considered to be not granted. This is partly blamed on missing semantics in the RELs. For example, ODRL has been criticised by some for the absence of a "*not*" semantic [7], which prevents rights holders from expressing a use license like "allow user B all rights except right A".

RELs allow for the expression of digital contracts, even though some, like Felten in [4], have argued that the RELs are unsuitable for expressing legal rights. However, contracts are usually negotiated between two parties, and true contracts require parties to communicate [5]. Referring to XrML, Mulligan et al. argued that "*the assumption of a one-way expression of rights has in part led to the current deficiencies in the REL*" [5]. Mulligan et al. concluded that a REL allowing bi-directional communication as well as *rights messaging protocols (RMP)* that support contract negotiations are essential in future DRM systems.

The problem with the current system can be best represented using an example from the second scenario in Microsoft's overview of RMS [2]. Tom creates a document for Jill, and protects it using RMS. He specifies that the document can only be viewed and edited by Jill for one week. If Jill requires additional time, Tom is required to edit the rights to the document, extend the deadline and then redistribute the document to Jill. However, this solution has some major drawbacks, like:

1) If the document in question is very big (presentation files for example can easily be over 50Mb in size), it may become impractical for Tom to redistribute the document every time rights need to be changed. Even with broadband Internet, many mail servers for example do not allow large attachments.
2) Tom could be out of the office, and thus may not necessarily be in a position to handle rights changes. If there are automated license servers, bi-directional RELs could allow end users to request for changes without the intervention of the rights holders.

With a bi-directional REL, it should allow the user and rights holder to conduct negotiations on the rights the user is given. This process can take more than a single round of "requests" to the rights holder and "offers" to the user. Furthermore, a bi-directional REL should also allow a user to request changes to an existing use license. Furthermore, a bi-directional REL potentially allows for upgrades to a use license after the initial issuing without the need to change the DRM controller or redistribute the protected data.

With bi-directional RELs it would also be possible to cater for fair use at a general level – rights holders can issue use licenses with usage rules fair for the majority of the users. If there are users who require additional privileges that fall under fair use (academics who would like to create extra copies for their lectures, journalists who would like to excerpt a quote for a review etc.), they can easily negotiate for these additional rules.

Electronic negotiation can be represented in a layered model as shown in figure 1. The users are involved in a *transaction*,

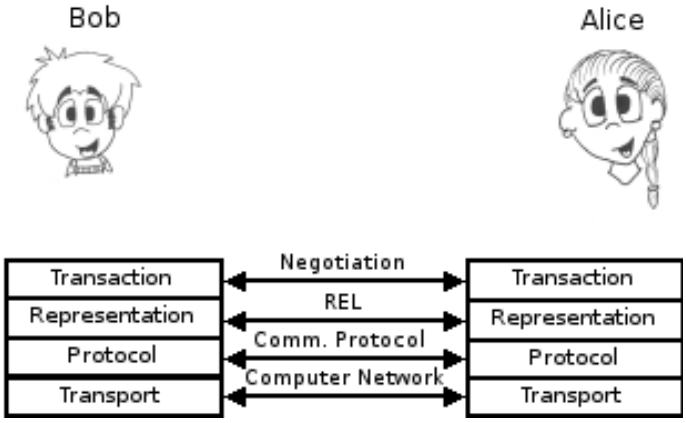


Fig. 1. Layered view of electronic negotiation

and in the case of contract negotiation, the contract will be a human readable contract. The contract is *represented* in a machine readable language, like ODRL. The negotiation takes place using a *communication protocol* over a *computer network*. Ideally, these layers should be independent, and thus the communication protocol should be separated from the REL. For this reason, this paper focuses on the ODRL extensions required to allow ODRL to express negotiations, and does not discuss the details of various negotiation protocols that could be used.

In this paper, we introduce vocabulary and syntax to facilitate bi-directional communication in ODRL. We motivate our design, detail individual elements and then provide two examples showing how a bi-directional ODRL can be used. We also detail a scenario, with examples, demonstrating the use of our extended language as a means for enabling fair use. Similar vocabulary and syntax can also apply to other RELs like XrML.

II. Design Motivations

In 2000, Park et al. [6] discussed the different distribution architectures that could be implemented for secure content distribution. Park et al. distinguished various architectures with three criteria: the presence of a virtual machine, the type of control set and the distribution style. They concluded that a virtual machine is required for secure content distribution, while the type of control sets and distribution style dictate the amount of control the "owner" of the content has after distribution. In a DRM system, the virtual machine represents the DRM controller and the control set represents the REL and the usage licence mechanisms.

Park et al. categorised control sets into three types: fixed control sets, embedded control sets and external control sets [6]. In *fixed control sets*, the DRM system comes with a predefined set of controls, and thus the DRM enabled data does not have to have any additional controls. In *embedded control sets*, the DRM enabled data comes with a set of controls as a single secure package while in *external control sets*, the control set and the DRM enabled data come in separate packages. It is possible to combine multiple type of control sets, as long as the DRM controller can regulate which control sets should be implemented; e.g. if the fixed control set does not allow copying, but the embedded control set (issued after the fixed control set) does allow copying then the DRM controller should allow copying.

To fully exploit the power of a bi-directional REL, the DRM system must allow for changes to be made to the protected work after distribution has taken place. Thus the DRM controller must be able to enforce all three types of control sets, and be able to handle use licenses that allow for rights previously disallowed.

It is true that any number of mechanisms can be used to express communication from the user to the rights holders. However, if the expression is not made in the language used by the rights holders to express rights, there will be a need to translate from the users' needs to the appropriate REL. Translation can be an expensive process, and can lead to ambiguities and inaccuracies. Thus having bi-directional support in a REL allows for the possibility of a standardised mechanism to express the needs of the end users.

In our design we envisage a bi-directional system to be implemented as a web-service. Thus a user would *request* changes to their current rights and can expect to receive three types of responses. Firstly, the rights holders can grant the request and issue a new license, which can be easily expressed with any REL. Alternatively, the rights holders can grant the request by creating a licence addendum (in a separate file) (*grant-request*). To handle this response, the DRM controller must be able to detect and use the extended license. Lastly, the rights holders can deny the request (*deny-request*). The user would need to be informed which requests are being denied since it may happen that the user requested three changes, of which only one is granted. Thus, in both the *grant-request* and *deny-request* there would be a need to include the requests.

There are three actions that a user could request:

- Request to **add** one or more permissions, resources etc. that are either not currently present or to extend the current values e.g. add one more week to the deadline
- Request to **remove** one or more permissions, resources etc. that have been granted through an earlier license or license addendum. While this feature is most probably not going to be in big demand, it could be used to strip down undesired or unused permissions. The remove feature is also necessary for:
- Request to **replace** one or more permissions that have been granted through an earlier license or license addendum. The request to replace is essentially a combination of an add and a remove request, but it would be more useful for tracking purposes to utilise a replace request mechanism. There should not be any restriction on how the replace mechanism is used – for example a user might request a replacement of dissimilar permissions, e.g. replace his right to print 5 copies with the right to make a backup.

With a bi-directional system, it would require the rights holders to keep track of individual licenses, and how the

licenses inter relate. The grant-request licenses should also be able to identify (possibly through the use of a URI) the original request as well as the original license. This would allow the DRM controller to keep track of the permissions, resources, etc. that have been removed or changed. For example, if the user originally had permission to print a document 2 times, printed it once, and then requested and received permission to print the document an additional 5 times, the DRM controller should allow the user to print 6 more times.

Lastly, we believe that the bi-directional extensions makes ODRL more *complete*. Current ODRL specifications allow for two types of licenses – an *offer* and an *agreement*. With an offer, the rights holders are allowed to express the rights that they are willing to offer to the end user. If the end user accepts, the rights holders can then create an agreement. With our extensions, it is now possible for the end user to have a more active part in generating the agreement, and thus allow for flexibility for the user.

In the following section, we discuss the details of our extensions.

III. ODRL-EXT: BI-DIRECTIONAL EXTENSIONS TO ODRL

Our extension adds three more entities – request, grant-request and deny-request – and are modelled on the agreement entity. We envisage its main use as being in a web-services environment and can be described in four easy steps. The end-user can request the rights holder for a set of rights on a set of assets. The rights holder can then evaluate the request, and then deny or grant that request. The user can accept the decision or carry on negotiating by refining his/her requests. This process is shown in figure 2

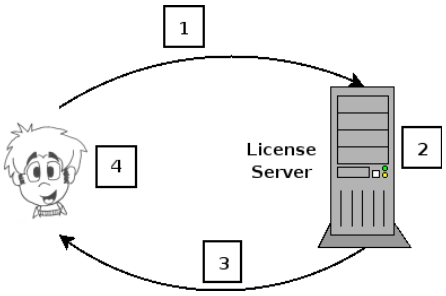


Fig. 2. Negotiating a use license

This model can be further extended where the rights holder can offer various rights at various prices. The prospective end user can then request a combination of rights, pay for these rights and then receive an end user license. Thus in this manner the request entity can be used for electronic contract negotiation. The grant and deny request entities can be used to conditionally accept or reject requests during the contract negotiation.

A. Add, Remove and Replace

The add, remove and replace requests are the base elements of our extensions. A user can request a combination of these requests, and similarly the rights holders can grant or deny the

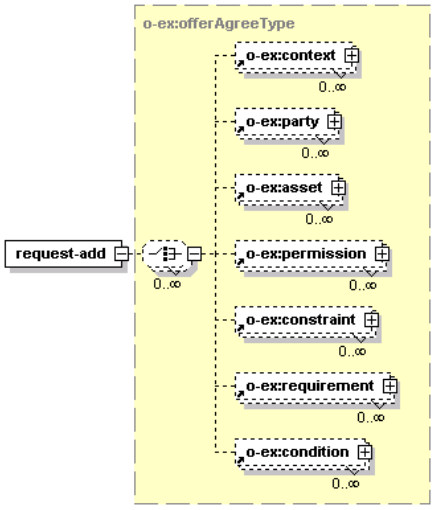


Fig. 3. The Add Request Content Model

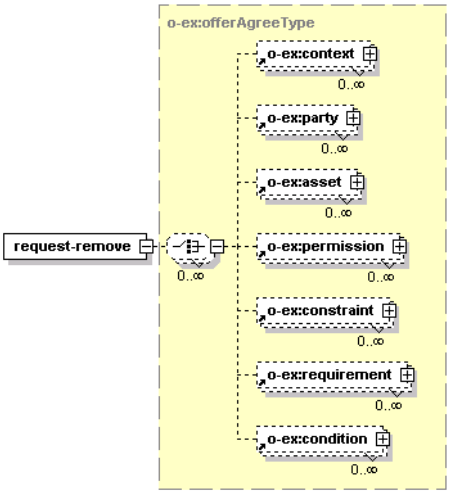


Fig. 4. The Remove Request Content Model

combination of the requests. For maximum flexibility, every element of a ODRL license agreement should be negotiable – permissions, constraints, requirements, conditions, assets and even the parties. For this reason, add, replace and remove elements are simply instances of the offerAgreeType in the ODRL Expression Language Schema [1]. Using the offerAgreeType also minimises ambiguity during negotiations, as the exact rights can be transfered to the "offer" license and eventually the "agreement" license.

The *replace-request* element comprises of a set of remove requests followed by a set of add requests. Although a replace-

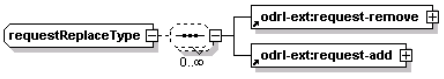


Fig. 5. The Replace Request Content Model

request element is not necessary, we believe that this element would allow for better tracking and management by the rights holders. This would also allow for automation of license servers, where the rights holders can write different rules on which combinations of replace requests they would allow. Figures 3,4 and 5 show the content model for the add, remove and replace elements.

B. Request

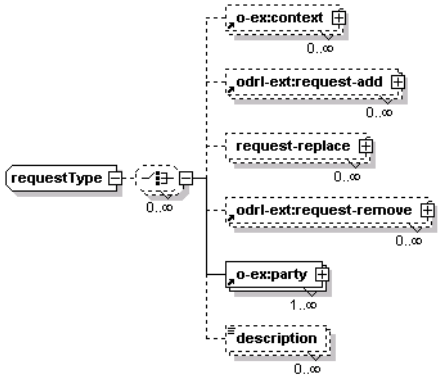


Fig. 6. The Request Content Model

The user communicates to the rights holders through a series of requests. The request element is the only element of the *requestType*. The requestType type, creates an envelope containing all the add, remove and replace requests from the user as well as the context of the request and information about the party making the request. The *context* element allows the rights holder to reconcile the request against an existing agreement or an offer. At least one party is required to identify the party making the request. The description element allows for the end user to write notes, and give more detailed information to the rights holder. If the request is processed manually, this feature can be very useful. Figure 6 shows the content model of the requestType.

C. Request Response

The *requestResponseType* creates an envelope for the rights holders to respond back to the user making the request.

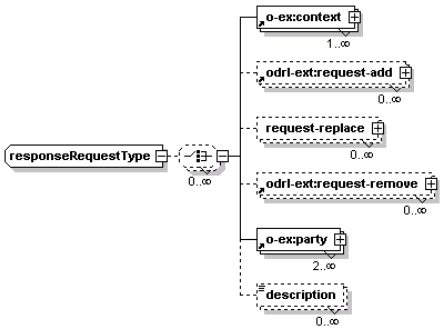


Fig. 7. The Response-Request Content Model

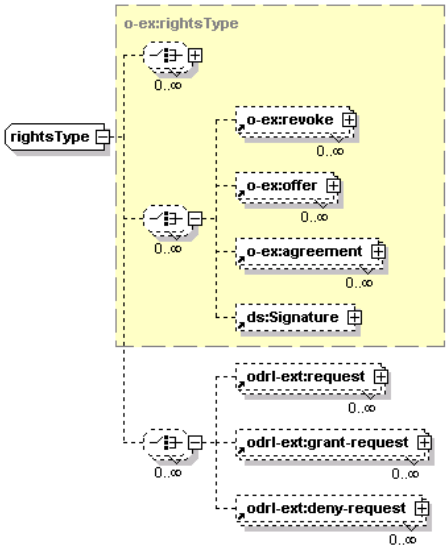


Fig. 8. The rightsType Content Model

There are two differences between the requestType and the requestResponseType. Firstly, the response from the rights holders must have a context, either of an earlier request or of the affected agreement. This will allow the DRM controller to keep track of the chain of agreements that it needs to manage and also allow the rights holders to track their responses to requests. Secondly the response must have at least two parties - one identifying the user who made the request and another to identify the rights holder. Figure 7 shows the content model of the requestResponse type. The rights holders can respond to a request from the end user in two ways – they can either grant or deny the requests, and thus the grant and deny request elements are of the requestResponseType.

D. rightsType

In ODRL 1.1 the *rightsType* complex type encapsulates agreements and offers with a digital signature and a revoke mechanism [1]. We extended this type to encapsulate the request, grant-request and deny-request elements.

We have also redefined the rights element to be of this type. Figure 8 shows the content model of the rights type. The rightsType in ODRL 1.1 extends the offerAgreeType and this portion has been collapsed in the diagram.

We recognise that these extensions could also be encapsulated in a new type (for example *negotiationType*) leaving the existing *rightsType* type alone. If this approach is taken, it would also need a digital signature and a revoke mechanism and we think that our current approach is more elegant as it avoids duplication of common functions.

E. Examples

In “Ebook Scenario #2” of the ODRL 1.1 specifications, a consumer (Mary Smith) purchases an ebook “Why Cats Sleep and We Don’t” [1]. The use license restricts consumers to a single CPU and allows them to print the book at most two times.

In example 1, the consumer requests the rights holders to be allowed to print the ebook 5 more times. Note, that for the sake of clarity we have left the namespace definitions and schema locations out of the example. The descriptions of the namespaces are detailed below.

odrl-ext: The extended ODRL schema as discussed in this section.

o-ex: The *Expression Language Schema* of the ODRL 1.1 specifications.

o-dd: The *Data Dictionary Schema* of the ODRL 1.1 specifications.

Example 2 shows a grant request should the rights holders grant the user's request. A deny request would be the same except the *grant-request* elements will be replaced with the *deny-request* element.

F. Full Listing

A full listing of the schema definition is available in the appendix .

```
<odrl-ext:rights>
  <odrl-ext:request>
    <o-ex:context>
      <o-dd:uid>urn:ebook.world/999999/
license/1234567890-ABCDEF</o-dd:uid>
    </o-ex:context>
    <odrl-ext:request-add>
      <o-ex:permission>
        <o-dd:print>
          <o-ex:constraint>
            <o-dd:count>5</o-dd:count>
          </o-ex:constraint>
        </o-dd:print>
      </o-ex:permission>
    </odrl-ext:request-add>
    <o-ex:party>
      <o-ex:context>
        <o-dd:uid>
          urn:ebook.world/999999/users/
msmth-000111
        </o-dd:uid>
        <o-dd:name>Mary Smith</o-dd:na
me>
      </o-ex:context>
    </o-ex:party>
  </odrl-ext:request>
</odrl-ext:rights>
```

Example 1: Simple ODRL Request

```
<odrl-ext:rights>
  <odrl-ext:grant-request>
    <o-ex:context>
      <o-dd:uid>urn:ebook.world/999999/
license/1234567890-GHIJKL</o-dd:uid>
    </o-ex:context>
    <o-ex:context>
      <o-dd:uid>urn:ebook.world/99999/
license/1234567890-ABCDEF</o-dd:uid>
    </o-ex:context>
    <odrl-ext:request-add>
      <o-ex:permission>
        <o-dd:print>
          <o-ex:constraint>
            <o-dd:count>5</o-dd:count>
          </o-ex:constraint>
        </o-dd:print>
      </o-ex:permission>
    </odrl-ext:request-add>
    <o-ex:party>
      <o-ex:context>
        <o-dd:uid>urn:ebook.world/99999
9/users/msmth-000111</o-dd:uid>
        <o-dd:name>Mary Smith</o-dd:na
me>
      </o-ex:context>
    </o-ex:party>
    <o-ex:party>
      <o-ex:context>
        <o-dd:uid>x500:c=AU;o=RightsDir
;cn=AddisonRossi</o-dd:uid>
      </o-ex:context>
    </o-ex:party>
    <o-ex:party>
      <o-ex:context>
        <o-dd:uid>x500:c=AU;o=RightsDir
;cn=EBooksRUS
        </o-dd:uid>
      </o-ex:context>
    </o-ex:party>
  </odrl-ext:grant-request>
</odrl-ext:rights>
```

Example 2: ODRL Grant Request

IV. Extended Example

Examples 1 and 2 used a simple scenario to demonstrate the use of our proposed extensions. In this section, we detail a more complicated scenario (based once again on "Ebook Scenario #2" in [1]) that also demonstrates how our extensions could be used as a means to enable fair use.

In the existing scenario, Mary Smith purchases an ebook "Why Cats Sleep and We Don't" [1]. Users are restricted to a single CPU and print the book at most 2 times (which we extended by another 5 copies in examples 1 and 2). Suppose, Mary Smith is a journalist and wishes to write a thorough review of the ebook and would like to excerpt some of the pictures for this purpose (excerption for the purpose of review is normally considered a fair use right). In this section, we detail the interactions between Mary Smith and the license server for this purpose.

Note, that for the sake of clarity we have left the namespace definitions and schema locations out of the example. The

descriptions of the namespaces are detailed below.

odrl-ext: The extended ODRL schema as discussed in this section.

o-ex: The *Expression Language Schema* of the ODRL 1.1 specifications.

o-dd: The *Data Dictionary Schema* of the ODRL 1.1 specifications.

o-dd-ext: An extension of the Data Dictionary Scheme of ODRL 1.1 to allow representation of credentials (discussed in sections IV-B and V).

A. Initial Request

Mary Smith wishes to excerpt 3 pictures from different pages in the ebook, the first picture in page 3 while the last picture is in page 56 (about half way through the book).

```
<odrl-ext:rights>
  <odrl-ext:request>
    <o-ex:context>
      <o-dd:uid>urn:ebook.world/999999/
license/1234567890-ABCDEF</o-dd:uid>
    </o-ex:context>
    <odrl-ext:request-add>
      <o-ex:permission>
        <o-dd:excerpt>
          <o-ex:constraint>
            <o-dd:range>
              <o-dd:min>3</o-dd:min>
              <o-dd:max>56</o-dd:max>
            </o-dd:range>
          </o-ex:constraint>
        </o-dd:excerpt>
      </o-ex:permission>
    </odrl-ext:request-add>
    <o-ex:party>
      <o-ex:context>
        <o-dd:uid>
          urn:ebook.world/999999/users/
msmth-000111
        </o-dd:uid>
        <o-dd:name>Mary Smith
        </o-dd:name>
      </o-ex:context>
    </o-ex:party>
  </odrl-ext:request>
</odrl-ext:rights>
```

Example 3: Extended Example – Request 1

B. Initial Rejection and Counter Offer

Excerption is a fair use, but is usually limited to a percentage of a work. The license server rejects Mary Smith's request with an explanation, but also offers a counter offer that could be used by Mary Smith. This counter offer makes use of a *credential* constraint not present in the standard ODRL data dictionary. The counter offer is given as a *grant-request*, although it could also be expressed as an *offer*.

```
<odrl-ext:rights>
  <odrl-ext:deny-request>
    <o-ex:context>
      <o-dd:uid>urn:ebook.world/999999/
license/TRANS-0101</o-dd:uid>
    </o-ex:context>
    <odrl-ext:request-add>
      <o-ex:permission>
        <o-dd:excerpt>
          <o-ex:constraint>
            <o-dd:range>
              <o-dd:min>3</o-dd:min>
              <o-dd:max>56</o-dd:max>
            </o-dd:range>
          </o-ex:constraint>
        </o-dd:excerpt>
      </o-ex:permission>
    </odrl-ext:request-add>
    <o-ex:party>
      <o-ex:context>
        <o-dd:uid>
          urn:ebook.world/999999/users/
msmth-000111
        </o-dd:uid>
        <o-dd:name>Mary Smith
        </o-dd:name>
      </o-ex:context>
    </o-ex:party>
    <odrl-ext:description>
Excerption is only available with an
academic, scholar or journalist
credential. Furthermore, a maximum of
10% of the total protected work can be
excerpted
    </odrl-ext:description>
</odrl-ext:deny-request>
</odrl-ext:rights>
```

Example 4: Extended Example – Response 1, the denial of request

```
<odrl-ext:rights>
  <odrl-ext:grant-request>
    <o-ex:context>
      <o-dd:uid>urn:ebook.world/999999/
license/1234567890-ABCDEF</o-dd:uid>
    </o-ex:context>
    <o-ex:context>
      <o-dd:uid>urn:ebook.world/999999/
license/1234567890-ABCDEF-01</o-dd:uid>
    </o-ex:context>
```

Example continued over the page

```
    <odrl-ext:request-add>
      <o-ex:permission>
        <o-dd:excerpt>
          <o-ex:constraint>
            <o-dd:range>
              <o-dd:min>3</o-dd:min>
              <o-dd:max>13</o-dd:max>
            </o-dd:range>
            <o-dd-ext:credential>
              <o-dd-ext:OrList>
            <o-dd-ext:CredentialsType>
                  Journalist
            </o-dd-ext:CredentialsType>
            <o-dd-ext:CredentialsType>
                  Academic
            </o-dd-ext:CredentialsType>
            <o-dd-ext:CredentialsType>
                  Scholar
            </o-dd-ext:CredentialsType>
              </o-dd-ext:OrList>
            </o-dd-ext:credential>
          </o-ex:constraint>
        </o-dd:excerpt>
      </o-ex:permission>
    </odrl-ext:request-add>
    <o-ex:party>
      <o-ex:context>
        <o-dd:uid>
          urn:ebook.world/999999/users/
msmth-000111
        </o-dd:uid>
        <o-dd:name>Mary Smith
        </o-dd:name>
      </o-ex:context>
    </o-ex:party>
  </odrl-ext:request>
</odrl-ext:rights>
```

Example 5: Extended Example – Response 2, A counter offer

C. Refined Request

Mary Smith decides to refine her request to suit the terms of the license server. She chooses to make a request to excerpt from three different parts of the book but with much smaller page ranges. She also decides to get the license specified for a “Journalist” credential only. The credential would form part of the protocol and not part of the negotiation message, and thus would be represented separately.

```
<odrl-ext:rights>
  <odrl-ext:grant-request>
    <o-ex:context>
      <o-dd:uid>urn:ebook.world/999999/
license/1234567890-ABCDEF-01</o-dd:uid>
    </o-ex:context>
    <odrl-ext:request-add>
      <o-ex:permission>
        <o-dd:excerpt>
          <o-ex:constraint>
            <o-dd:range>
              <o-dd:min>3</o-dd:min>
              <o-dd:max>4</o-dd:max>
            </o-dd:range>
            <o-dd:range>
              <o-dd:min>16</o-dd:min>
              <o-dd:max>18</o-dd:max>
            </o-dd:range>
            <o-dd:range>
              <o-dd:min>56</o-dd:min>
              <o-dd:max>57</o-dd:max>
            </o-dd:range>
            <o-dd-ext:credential>
              <o-dd-ext:CredentialsTy
pe>
                  Journalist
              </o-dd-ext:CredentialsTy
pe>
            </o-dd-ext:credential>
          </o-ex:constraint>
        </o-dd:excerpt>
      </o-ex:permission>
    </odrl-ext:request-add>
    <o-ex:party>
      <o-ex:context>
        <o-dd:uid>
          urn:ebook.world/999999/users/
msmth-000111
        </o-dd:uid>
        <o-dd:name>Mary Smith
        </o-dd:name>
      </o-ex:context>
    </o-ex:party>
  </odrl-ext:request>
</odrl-ext:rights>
```

Example 6: Extended Example – Request 2, A refined request

D. Accepted Response

The license server accepts Mary Smith’s request and issues a grant request use license.

```
<odrl-ext:rights>
  <odrl-ext:grant-request>
    <o-ex:context>
      <o-dd:uid>urn:ebook.world/999999/
license/1234567890-ABCDEF</o-dd:uid>
    </o-ex:context>
    <o-ex:context>
      <o-dd:uid>urn:ebook.world/999999/
license/1234567890-ABCDEF-01</o-dd:uid>
    </o-ex:context>
    <odrl-ext:request-add>
      <o-ex:permission>
        <o-dd:excerpt>
          <o-ex:constraint>
            <o-dd:range>
              <o-dd:min>3</o-dd:min>
              <o-dd:max>4</o-dd:max>
            </o-dd:range>
            <o-dd:range>
              <o-dd:min>16</o-dd:min>
              <o-dd:max>18</o-dd:max>
            </o-dd:range>
            <o-dd:range>
              <o-dd:min>56</o-dd:min>
              <o-dd:max>57</o-dd:max>
            </o-dd:range>
            <o-dd-ext:credential>
              <o-dd-ext:CredentialsTy
pe>
                  Journalist
              </o-dd-ext:CredentialsTy
pe>
            </o-dd-ext:credential>
          </o-ex:constraint>
        </o-dd:excerpt>
      </o-ex:permission>
    </odrl-ext:request-add>
    <o-ex:party>
      <o-ex:context>
        <o-dd:uid>
          urn:ebook.world/999999/users/
msmth-000111
        </o-dd:uid>
        <o-dd:name>Mary Smith
        </o-dd:name>
      </o-ex:context>
    </o-ex:party>
  </odrl-ext:request>
</odrl-ext:rights>
```

Example 7: Extended Example – Response 3

V. Future Work

As pointed out by Mulligan et al. [5], bi-directional communication does not depend on REL support only. The protocols used by the DRM systems and the DRM controllers need to be modified to allow for bi-directional communication and for better management of multiple use licenses for the same digital object.

License servers could also be setup to grant or deny certain requests automatically, and thus algorithms are needed to automatically evaluate license templates (ODRL offers) against user requests.

We are currently investigating the use of credentials in DRM, particularly as a mechanism in allowing for fair use (as shown in section IV). Together with a bi-directional REL, we believe that most of the common fair uses can be accommodated in a DRM system.

In the broader scheme, bi-directional REL forms a core part of our proposal to create an open right management services framework [3], and will hopefully overcome many of the current obstacles in DRM systems. A smaller sub-project is currently implementing some of the extensions for DRM controllers mentioned above.

VI. Conclusions

In this paper we discussed extensions to ODRL to allow for bi-directional communication. We discussed our motivation, the concept model, the syntax and semantics of the extensions. Furthermore, we presented examples using existing ODRL scenarios that make use of our extensions. Finally, we discussed how the extensions could be used to allow for fair use with examples drawn from an existing ODRL scenario.

The extensions allow end users to specify any part of a use license including rights, constraints and resources, they would like to have in a use license and rights holders to respond to these requests, thus allowing for negotiations of rights. These extensions complement the existing “offer” and “agreement” license types, and make ODRL more complete.

By extending the XML schema, we have not broken the existing standard; and thus allows for full backward compatibility. We believe that the request feedback mechanism would allow for easier rights management through better contract negotiation, and would also allow for users to request (and be subsequently granted) fair use rights that might not necessarily hold for everyone. The extensions we have presented can also be implemented in other RELs such as XrML.

Acknowledgements

This work is partially supported through grants from the UCT Council and the National Research Foundation (NRF) of South Africa. Any opinions, findings, and conclusions or recommendations expressed in this paper/report are those of the author(s) and do not necessarily reflect the views of UCT, the NRF or the trustees of the UCT Council.

XML Schema content model diagrams were generated using XMLspy.

Alice and Bob created by Nicholas Hall.

REFERENCES

[1] *Open Digital Rights Language (ODRL) 1.1*, 2002, URL: http://odrl.net/1.1/ODRL-11.pdf.

[2] "Technical overview of windows rights management services for windows server 2003," Microsoft," White Paper, 2003.

[3] A. Arnab and A. Hutchison, "Distributed DRM System," University of Cape Town," Departmental Technical Report, No. CS04-27-00, 2004.

[4] E. Felten, "Skeptical view of DRM and Fair Use," *Communications of the ACM*, vol. 46, no. 4, pp. 57–59, 2003.

[5] D. Mulligan and A. Burstein, "Implementing Copyright Limitations in Right Expression Languages," in *Proceedings of the 2002 ACM workshop on Digital Rights Management*. ACM, 2002.

[6] J. Park, R. Sandhu, and J. Schifalacqua, "Security architectures for controlled digital information dissemination," in *Proceedings of the 16th Annual Computer Security Applications Conference*, 2000.

[7] R. Wenning, "DRM and the Web," in *ODRL International Workshop 2004, Vienna Austria*, 2004, URL: http://www.w3.org/Talks/2004/04-odrl/.

APPENDIX

In this section, we provide a full source listing of the extended ODRL schema. Due to space constraints, indentation has been reduced.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://people.
cs.uct.ac.za/~aarnab-ODRL"
elementFormDefault="qualified"
attributeFormDefault="qualified" version="
0.1" xmlns:odrl-ext="http://people.cs.uct.
ac.za/~aarnab-ODRL"
xmlns:xs="http://www.w3.org/2001/XMLSchema
" xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
>

<xs:import namespace="http://odrl.net/1.1/
ODRL-EX" schemaLocation="http://www.odrl.n
et/1.1/ODRL-EX-11.xsd"/>

<xs:annotation>
 <xs:documentation>
   XML Schema extends ODRL Expression Lang
   uage Schema by allowing users/distribut
   ers to requ-est rights from theright ho
   lder.

   Alapan Arnab
   Validated with XMLSpy 2004
 </xs:documentation>
</xs:annotation>


<xs:element name="rights" type="odrl-ext:
rightsType"/>

<!-- Add the query element to the language
-->
<xs:element name="request" type="odrl-ext:
requestType"/>
<xs:element name="grant-request" type="odr
l-ext:responseRequestType"/>
<xs:element name="deny-request" type="odrl
-ext:responseRequestType"/>


<!-- The request type comprises of a numbe
r of addition, replace and remove requests
. These requests themselves are of the off
erAgreeType.
-->

<xs:complexType name="requestType">
 <xs:choice minOccurs="0" maxOccurs="unbou
nded">
  <xs:element ref="o-ex:context" minOccurs
="0"
   maxOccurs="unbounded"/>
  <xs:element ref="odrl-ext:request-add"
   minOccurs="0" maxOccurs="unbounded"/>
  <xs:element name="request-replace"
   type="odrl-ext:requestReplaceType"
   minOccurs="0" maxOccurs="unbounded"/>
  <xs:element ref="odrl-ext:request-remove
"
   minOccurs="0" maxOccurs="unbounded"/>
  <xs:element ref="o-ex:party"
   maxOccurs="unbounded"/>
  <xs:element name="description" type="xs:
string"
   minOccurs="0" maxOccurs="unbounded"/>
 </xs:choice>
</xs:complexType>


<!-- A grant/deny request should have the
information about the request its granting
, the license number/context information
of the original request and license.contex
t information about the new license.-->

<xs:complexType name="responseRequestType"
>
 <xs:complexContent>
  <xs:restriction base="odrl-ext:requestTy
pe">
   <xs:choice minOccurs="0" maxOccurs=
    "unbounded">
    <xs:element ref="o-ex:context"
     maxOccurs="unbounded"/>
    <xs:element ref="odrl-ext:request-add"
     minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="request-replace"
     type="odrl-ext:requestReplaceType"
     minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="odrl-ext:request-remo
```

```
ve"
     minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="o-ex:party" minOccurs
="2"
     maxOccurs="unbounded"/>
    <xs:element name="description"
     type="xs:string" minOccurs="0"
     maxOccurs="unbounded"/>
   </xs:choice>
  </xs:restriction>
 </xs:complexContent>
</xs:complexType>

<!-- Allows for a multiple number of tuppl
es for replacement.-->

<xs:complexType name="requestReplaceType">
 <xs:sequence minOccurs="0"
  maxOccurs="unbounded">
  <xs:element ref="odrl-ext:request-remove
"/>
  <xs:element ref="odrl-ext:request-add"/>
 </xs:sequence>
</xs:complexType>

<xs:element name="request-add"
 type="o-ex:offerAgreeType"/>

<xs:element name="request-remove"
 type="o-ex:offerAgreeType"/>

<!-- The rightType container. Added the re
quest container. -->

<xs:complexType name="rightsType">
 <xs:complexContent>
  <xs:extension base="o-ex:rightsType">
   <xs:choice minOccurs="0"
    maxOccurs="unbounded">
    <xs:element ref="odrl-ext:request"
     minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="odrl-ext:grant-reques
t" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="odrl-ext:deny-request
" minOccurs="0" maxOccurs="unbounded"/>
   </xs:choice>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
</xs:schema>
```

Predicting the evolution of digital rights, digital objects, and digital rights management languages.

Jonathan Schull, Associate Professor, Information Technology, Rochester Institute of Technology

***Abstract*— As a sometime biological psychologist and sometime DRM pioneer, I suggest that biological principles are at least as important as technological principles in anticipating future developments in the field of rights management, and requirements for digital rights languages. Among those possible developments are (1) increases in the virtuality and virality of rights-managed objects, of distribution systems, and of payment systems, (2) systems for tracking the copying and redistribution of digital documents, (3) application of digital rights to data derived from document tracking, (4) attribution of those rights to the individuals who do the re-distributing, (5) development of rights management systems for the aggregation, protection, anonymization, and monetization of personal information, (6) rights-managed digital objects whose content changes spontaneously as a function of normal use, and (7) digital objects that adapt through a natural selection-like process of mutation, recombination and differential reproduction.**

Such ideas pose interesting challenges for rights management languages.

***Index Terms*—Copyright Protection, Rights Management, Superdistribution, Natural Selection**

I. Introduction

ANY rights management language that hopes to keep pace with "facts on the ground" must be extensible to rights management practices that are uncommon, but predictable, today. We can better design a digital rights language for the future if we can anticipate the changes and change processes we will have to accommodate.

As a sometime biological psychologist and sometime DRM pioneer [1,2,3], I believe that biological principles are at least as important as technological principles in anticipating future developments in the field of rights management.

Today's digital rights management situation represents the convergence of two historical trends: virtualization and biologization. Of the two, biologization is the least discussed, let alone well understood. But its implications are most fundamental for digital rights management and for the transformation of the information economy.

Manuscript received June 13, 2005.

Author is with the Interactive Media Group in the Information Technology Department at the Rochester Institute of Technology, Rochester New York 14607. Phone: 585-738-6696. email schull@digitalgoods.com

Here's what I mean. Traditional economies are based upon the delivery of valuable “things” (products and services) in exchange for receipt of valuable “things” (including money). However, starting at the dawn of civilization, value came to be represented first by tokens, then by coin, then symbolically in money, and then virtually in disembodied bits. That's virtualization: symbolic representation with more and more impact with less and less mass and energy.

At the dawn of life, value was embodied first in analog form in the biological processes of single celled organisms, and then symbolically in digital form by RNA and DNA. That too is virtualization. But life also teaches us that when things are virtualized, reproduction becomes easier, and biological dynamics of reproduction and evolution arise. Virtualization is a step on the road to biologization.

Today, information products are being virtualized. To fully understand alternative rights management options, it may be helpful to look closely at a spectacularly successful economy based not on state-sanctioned currencies, but upon unregulated reproduction, competition, and innovation. That economy is all around. It is the world of biology.

My goal in this paper will be to provide a broad historical, if idiosyncratic perspective, on the past and present evolution of digital objects and rights management systems. Needless to say, these ideas are offered as useful speculations, not confident predictions, about the future.

II. Behavioral engineering and Digital Goods

In the early 1990s I was a biological psychologist and amateur programmer interested in the co-evolution of biological, social and informational ecologies [4,5]. I studied animal behavior, and had created some useful software for analyzing my data. I wanted to distribute this software, to be compensated for my work, and to take advantage of the then-emerging virtualization of software products by distributing and selling my software over the Internet. I wanted my software to reproduce, like a positive virus, so that users would "infect" their friends by making and sending copies. In those days I was literally studying and observing paramecia as they swam around, reproduced, and proliferated; I had those images in my mind.

The shareware concept had been around for almost a decade, [6] but I knew that my customers, like me, were unlikely to assemble a check, an envelope, and a stamp if their only

reward was the delivery, weeks later, of a now-redundant diskette or a postcard-of -thanks. As a behaviorist, I knew that contingencies of reinforcement dictate that meaningful and relevant rewards should be delivered within a fraction of a second of the behaviors they are intended to encourage.

The instant reward I could use was obvious—increased access to the most valuable features of my product. But it was less obvious how I could ensure that the product would be purchased again (and again) each time it was redistributed. Before we consider that puzzle, I'd like to revisit the twin issues of virtualization and biologization. Because it turns out that my solution to this practical problem also led me to rethink my understanding of the information economy.

III. VIRTUALIZATION AND ECONOMICS

Even with shareware, money usually changes hands under the consensual delusion (or user-interface metaphor) that information products are things-- "goods"—and that publishing is a business in which manufactured things (like books) are traded for things (like gold doubloons) owned by the purchaser. The irony, of course, is that what consumers "hand" over these days typically cannot be "handled"--it is symbols (digitally encoded, perhaps in plastic credit), which give the "bearer" (who "bears" nothing) the right to control the disposition of other symbols in the future. And what consumers get back from publishers is less and less likely to be physical as well: software and music, books and movies are all moving into a realm in which delivery and consumption is the symbolically-controlled execution of virtual operations by virtual machines in virtual places "on the web" or "in the bank".

As these examples show, money was virtualized long before other forms of intellectual property. And as money became virtualized it became more and more copyable. Today it takes the constant vigilance and full force of the most powerful political and military forces in the history of mankind--governments, businesses, and the police forces that back them—to prevent money from being copied by unauthorized parties. Thus, copy-prevention is a time-honored solution to the fact that virtual goods are copyable goods.

However, while copy prevention may well be necessary to preserve the integrity of our monetary system and civilization, as we know it, it may well be counter-productive when it comes to other virtual value-objects. The thing-based transaction-metaphor adopted by commercial publishing may have outlived its usefulness.

IV. VIRTUALIZATION, BIOLOGIZATION AND THE INFORMATION ECONOMY

"Publishing" actually has two very different meanings and histories. The “thing-based manufacturing metaphor” can be said to have started with Gutenberg: books are manufactured, and exchanged for “cash on the barrelhead”. But there is also a much-older idea-based information dissemination activity called “publishing” that has been practiced non-commercially for millennia by authors, scholars, pamphleteers, theologians, by flowers (which disseminate vast amounts of genetic information and arrange to have it distributed, at little cost, on the wings of the wind.)

The essential “product” in this case, is information. And information is not a thing. It is a process by which patterns "in-form"--impress themselves upon--things. Furthermore, as we have noted, because these patterns are only loosely coupled to the media they inform, they reproduce, they spread, and they evolve. They don’t just move from place to place like traditional “things”.

To make a long story short, patterns that reproduce, spread and evolve originated in the primal soup 3-4 billion years ago, they spread into (and helped create) protocells, RNA, DNA, and organisms that make their living by in-forming their environment. Approximately 1 billion years ago, propagating patterns branched out to a new media-- animal nervous systems--that allowed them to reproduce, first via learning, then via spoken patterns of sound, then via written patterns of ink on paper, and just in the last century, as patterns of electrons in yet another culture-medium that is now known as the global internet. [7,8]

Thus, over the last century the remarkable dynamics and “technology” of biology have come to be understood. My claim is that digital rights practitioners need to recognize that those dynamics and emerging analogous technologies are an increasingly fundamental part of their own discipline.

V. NATURE'S PUBLISHING ECONOMY

The "economy of nature" depends relatively little on the principles of thing-based manufacturing economies. Plants and animals do sometimes organize reciprocal resource exchange relationships, but the resources that are exchanged are services (including reproductive services) as often than as they are things. Here's how this observation applied to my own work, and the concept of superdistribution.

You will recall that I wanted my users to copy and redistribute my software, and I wanted to be able to reward those who decided to purchase it by giving them instant access to the product's advanced features. I imagined a happy purchaser passing a copy on to a friend with a recommendation. When the friend executed the program, she would have limited access to the advanced features her friend had purchased, until she committed to a purchase. The moment she made a payment the product would provide full access. However she passed copies on to her friends the copies needed to revert to “demo” mode. Thus, I needed a lock that would respond to a combination of code plus context.

I'm sure there were other ways of getting to the right answer, but my inspiration was biology. Biological functions are embodied not in genes nor in the environment, but in the dynamic interaction of genes (code) and the environment. Change either genes or environment, and function (skin color, say) may change.

My code was not going to change; it was going to be copied perfectly (and, I hoped, often). But the environment of one user would be different from the environment of another user. So I could have my code behave differently when it detected

that it had been moved from the environment of a purchaser to the environment of a non-purchaser.

(In most systems, including mine, the environment that the software responds to is the user's computers. But in the patent I eventually wrote, and in the future, the enabling environment should be the user herself. After all, it is she who purchases the service. Rights management languages are going to have to accommodate the vagaries and constraints of biometric systems. *Can matters of biology and individuality be expressed in ODRL?)*

So here is how I ended up vending my animal behavior software. When the program started up, it profiled the user's computer, made a list of relatively stable but idiosyncratic features, added up all the ASCII values of the characters in that list (literally!) to produce a large number, used that computer "fingerprint" as the seed to a random number generator, and generated a many-digit magic "password". The program then looked for that magic number in a "password file" on the users hard drive, and if the right number could be found in that password file, it functioned in "professional mode"; if not, it functioned in "demo mode" and encouraged the user to by his own password. The nice thing about this arrangement was that even if the password file was copied and redistributed along with the software, the program would still come up in "demo mode" because the magic password for one user's machine was not valid for another user's machine.

Now, the only person who knew how to generate passwords was me. When a customer decided to purchase, she called a software vendor (by phone), he took payment (by credit card or purchase order) and wrote down the fingerprint, and he called me (by phone). I would get calls (sometimes while delivering lectures on cultural transmission and gene environment interaction) and speak the password to the vendor who would later speak it to the customer who would later type it into her password file.

After a year of this, I realized (1) this was working (2) the idea was potentially more significant (even as biology!) than the animal behavior I was trying to analyze (3) that it could be applied to software products other than mine (4) that software was a service (even though I occasionally referred to my business as a random number manufacturing and vending facility) (5) a password vending service was a good job for a computer--running from class to phone to computer to phone and back to class was silly. So the patent I wrote [1] and the business I started was (SoftLock Services *aka* DigitalGoods.com) was based on the idea of a software toolkit that could accommodate multiple authors, multiple products and multiple features, all coupling these product to a password vending system that took payments, delivered passwords, and distributed funds to software developers (and us).

VI. The history of "Superdistribution"

To my knowledge this is the earliest example of software-only "superdistribution". The term itself was invented some years earlier by a Japanese computer scientist named Ryoichi Mori who defined it as an "approach to distributing software in which software is made available freely and without restriction but is protected from modifications and modes of usage not authorized by its vendor"[9] But in fact, Mori's own system presumed the existence of special tamper-proof hardware, as did Brad Cox who popularized the concept and emphasized usage-metering in book and magazine publications around 1994[10,11] The concept was further popularized, and arguably co-opted, by Intertrust's founder Vincent Shear.[c.f. 12]

My impression is that most people think of "superdistribution" as a software-only process, like what I implemented. But in any case a software-only process is certainly more virtual and more viral than one that requires the distribution of special hardware

It's worth noting, however, that today's superdistribution concepts can be taken still further. Superdistribution could be more virtual. We don't have to assume an "earthbound" payment processing system run by a credit card processing system and linked to the banking network. With peer to peer architectures and web services, its possible to imagine a system in which software services or non-monetary information assets were the only "coin of the realm," with transactions being remunerated not with money but with scrip, redeemable for services or information assets. While some of these services would presumably have to be redeemable somewhere, somehow, for something of "nutritional" or "reproductive" value, our concept of payments as well as our products can and will go ever more virtual. *It's not clear to me whether ODRL can currently accommodate non-financial remuneration.*

A well-worked out example of non-monetary currency is "whuffie," as described in digital rights activist Cory Doctorow's science fiction novel, Down and Out in the Magic Kingdom [13] which depicts a world in which "whuffie" an constantly updated measure of reputation that motivates people to do useful and creative things. Anything is available to you if you have good whuffie, and those who make those goods available gain whuffie indirectly. But if you make a lot of enemies, your whuffie plummets. It's a good read, and except for the fact that the whuffie market is mediated by internet-connected brain implants, this futuristic scenario is actually hundreds of millions of years old: among many social mammals mating opportunities and access to environmental resources often based upon hard-earned social status.

A less outlandish example of non-monetary currencies arises when we consider compensating users for virally superdistributing content. Consumers who recommend and distribute products to their friends are providing marketing, distribution, sales, and technical support services to their recipients. Why should they not be compensated? And if we are going to compensate them, why not compensate them with something that we can "manufacture" at no cost—the right to consume other digital products?

Rights management languages will therefore face new challenges as the virtuality and virality of superdistribution arrangements increases. *Can ODRL specify compensation rights for people who redistribute but do not modify rights-managed content, and can it specify alternative currencies?*

(Incidentally during the "Great Ebook Boom of March, 2000", when Stephen King's published his ebook "Riding the

Bullet [14], I tried to determine how much redistribution was actually happening. To my surprise and dismay, there was relatively little. A survey suggested the reason--many of our customers told us they thought that that "wasn't allowed", even though our marketing materials explicitly encouraged them to pass copies to their friends. So one reason we were interested in compensating redistributors was to create some pro-copying propaganda to counter industry brainwashing that implies, with misleading simplicity, that copying violates copyrights.)

VII. Tracking information flows?

In order to compensate users for redistributing our products, we would need a good way of tracking redistribution. As a would-be "information ecologist" this was of great interest to me for other reasons as well.

First, I think that tracking the flow of digital objects and activities is a huge scientific opportunity. A field biologist once told me that hydrologists sometimes map Biscayne Bay in Florida by dropping thousands of oranges into the water, and taking aerial photos a day later. Because oranges float just beneath the surface and drift with the currents, the aerial photos capture a huge "map" marked out in orange-dotted lines. The lines trace water currents; interruptions in the lines show shipping lanes, deviations in the lines provide clues to submerged topographies, and so on. The shapeless murk of Biscayne Bay is illuminated and articulated simply by tracking the flow of waterborne objects through the system.

We live in a transparent, sea of cyberspace, and for the first time in history the flow of information through that sea is trackable and accessible over a global, growing Internet. This is a major development in the multi-billion year history of life and mind, and it is happening in our lifetimes. It is a big story, and a big scientific opportunity.

Second, as a sometime entrepreneur I think that tracking documents and information transactions will be a big business opportunity. When the information economy is as significant as the physical economy, "infonomic indicators" should be as important and as valuable to economists and market analysts as balance of trade statistics, the Dow Jones Industrial Index, etc. etc.

This raises further questions for rights-language developers. *Can ODRL allow content owners or superdistributors to claim ownership of valuable tracking data that are by-products of data-transactions, but not embodied in the rights-managed digital object itself?*

VIII. Tracking information flows: how?

Not surprisingly, my ideas about how to track information flows came from biology [2]. By exploiting the fact that each individuals genetic code is unique yet similar to that of close relatives, biologists have recently learned to reconstruct amazingly precise lineages of descent (pedigrees) going back hundreds of generations. These techniques have produced profound advances in biology, ecology, medicine, pharmaceutics, forensics, etc. Similarly significant advances would probably follow from a comparable system for reconstructing digital pedigrees of redistributed and evolving digital objects. After all, digital objects are increasingly the DNA of civilization.

One way to make digital objects trackable is record document transformations, reproductions, and the current context of use in a data field embedded within the object. Each time the object is accessed, we can check to see if the current context matches a previously stored fingerprint of the context, and if it does not, we can know that the object has been moved to a new context. In that case, we can append the new context fingerprint to the data field (thus preserving lineage information) and update our record of current context. In this way (and there are other ways) each digital object could have a family tree that would allow us to trace redistributed objects back, through all of its intermediary stages and users, to the original source. Then we can examine those data objects "in the field," or monitor their passage through mail servers, or have them periodically "phone home" to databases, and cross-reference with other data about purchases, purchasers, etc.

Does ODRL allow us to assert rights over, and prohibit tampering of, portions of a document that are intended to change, randomly or in a directed fashion, over time?

IX. Redistribution and the Privacy Problem

Document tracking also raises profound ethical issues. I suspect that a lot of redistribution tracking is already happening, but that it is unpublicized because document tracking invades the privacy of those who receive files as well as those who send them. It's a serious concern--suppose right-to-lifers used this methodology to identify and harass women to whom friends forward documents on abortion counseling? And recommended best practices are of little help: even if senders are informed about corporate privacy policies and allowed to specify the uses to which their personal data might be put, recipients of redistributed documents have no such choice or control.

I think digital rights management languages could be pressed into service here. Since the privacy problem has become a digital data problem, why not treat personal data as intellectual property owned by the people to whom it applies? If each us owned our personal data, each of us could use rights languages, copyright laws, and rights management systems to protect our privacy, fatten our wallets, and/or heighten public awareness of intellectual property law. The masses would benefit from the growing power of intellectual property law, and we could encourage people to make valuable data available and marketable. *(Does ODRL allow users to assert ownership of data generated by their handling of a given document?)*

A number of organizations have envisioned an anonymizing infomediary service, a "Personal Information Trust" (PIT), which collects, protects, and optionally sells anonymized personal information data in such a way that marketers could communicate with specific individuals (with consent under specified conditions) without learning the individual's identities, and in such a way that each individual could discontinue that communication at any time. Essentially the PIT would be a "go-between" or "Swiss bank account" that

could increase the value, and decrease the liabilities, of personal information by pooling information from diverse sources and by making a market for information buyers and sellers. [15]

The economics and ecology of the PIT would be quite interesting, because isolated snippets of data become more informative and therefore more valuable when they are commingled with other data in the PIT. This would allow the PIT to pay information deposits, and each information purchase would be add still more value to the PIT because data the information purchases are themselves valuable. A healthy PIT, like a healthy ecology, could actually "clean" the personal information environment by creating a value-gradient that would cause personal information to aggregate in the value-enhancing, privacy protecting, database, where it would earn money for the PIT and for the people it represents, through the sale of data and permission to contact targeted consumers.

While one can imagine many models for the governance, economics, and regulation of the PIT, the initial questions for rights language developers are clear. *Can ODRL be applied to data generated by information transactions between and among individuals, marketers? Does it allow individuals to specify the conditions under which they are willing to be contacted by marketers, or to let marketers or analysts make use their personal data (anonymized or otherwise) for other purposes?*

X. Evolving Digital Life

As I said earlier, living things don't just move from place to place. They reproduce, they spread, and they evolve. So far I have argued that digital objects can reproduce, can spread, and can be profitably tracked much like living things. I now want to suggest that its just a matter of time before they are "genetically engineered" to evolve and adapt through a process very much like natural selection.

Consider the case of a computer program that runs in "demo mode" for a certain number of minutes before demanding that the user purchase a "professional license". What is the right number of minutes? This might be hard to predict, and might vary from one market niche to another. But (1) if the number of minutes is controlled by a mutatable data field, and if (2) the number of minutes influences the probability that users will copy and redistribute the product, then the number of minutes should evolve, through random mutation and differential reproduction, toward values that maximize the likelihood of redistribution.

Thus, by putting functional aspects of a digital object under the control of mutatable code embedded in a frequently copied object, the conditions for natural selection could be created. We would want to select functional aspects that might affect the utility or attractiveness of a product, and we would want to constrain the degree of functional variation so that mutations could not have unacceptably negative (or fatal) effects. But even within such constraints there are many ways we might do this.

Of course, natural selection maximizes reproduction and this may not maximize purchasing, which is what product creators probably care about. But there are ways in which we might select mutations for purchase-encouragement rather than for copy-encouragement per se (see [2], columns 11 and 12).

The point is that in the long run, the difference between software and biology may become vanishingly small. Differential reproduction of inheritable characteristics – may eventually become another tool in the toolbox of the software engineer and the information marketer. If and when that happens, a new chapter in the billion-year history of life and life-like evolution may have begun. Indeed, in retrospect, we may conclude that the new chapter has already begun.

The last few decades brought us several digital revolutions, the open source software movement, the spam explosion, the copyright and patenting of DNA sequences, genetic algorithms, the onslaught of computer viruses and worms, and the emergence of a global information network. All of these things are driven by the "out of control" replication and propagation and evolution of digital objects, many with significant commercial value and social significance. It is the presumptive function of rights management languages to describe and facilitate the regulation or husbandry of these phenomena.

In this sense, rights management languages are themselves among the most interesting recent developments in the primal soup that constitutes today's information ecology. It will be interesting to see how well rights management languages can be designed for adaptive evolution.

References

[1] Schull, J. Method for encouraging purchase of executable and non-executable software. US Patent 5509070, 1992.

[2] Schull, J. Method for tracking software lineage. US Patent 6266654, 2001.

[3] Open Ebook Forum, Framework for an E-publishing Ecology, v0.78, 2000. www.openebook.org/doc_library/ecology/A%20Framework%20for%20the%20Epublishing%20Ecology.pdf

[4] Rozin, P., Schull, J. Adaptive-evolutionary perspectives and experimental psychology. In S.S. Stevens' Handbook of Experimental Psychology, R. Atkinson, R., Hernstein, G. Lindzey, & R.D. Luce (Eds.), New York: Wiley, 1988

[5] Schull, J. The View from the Adaptive Landscape. In Parallel Problem Solving from Nature, H.P. Schwefel and R. Manner, eds. Springer Verlag, 1991.

[6] Association of Shareware Professionals. History of Shareware. http://www.asp-shareware.org/users/history-of-shareware.asp 2000.

[7] Dawkins, R. The Selfish Gene. Oxford: Oxford University Press, 1976. Chapter 11.

[8] Kelly, K. Out of Control: The New Biology of Machines, Social Systems, and the Economic World, 1994. Reading, Mass.: Addison-Wesley, 1994. http://www.kk.org/outofcontrol/contents.php

[9] R. Mori, and M. Kawahara, Superdistribution: The Concept and the Architecture. THE TRANSACTIONS OF THE IEICE; VOL.E 73, NO.7 JULY l990 Special Issue on Cryptography and Information Security. http://www.virtualschool.edu/mon/ElectronicProperty/MoriSuperdist.html#Mori

[10] Cox, B. Superdistribution. *Wired*, 2.09, Sept 1994. http://www.wired.com/wired/archive/2.09/superdis.html

[11] Cox, B. *Superdistribution*. Reading, Mass.: Addison-Wesley, 1995

[12] Weber, R. The Superdistribution Chronicles, Pt. 1. http://robertweber.typepad.com/rightsmanagement/2005/02/the_superdistri.html

[13] Doctorow, C. *Down and Out in the Magic Kingdom.* New York: Tor, 2003. http://www.craphound.com/down/download.php

[14] King, S. Riding the Bullet.
http://www.simonsays.com/content/content.cfm?sid=33&pid=479688
[15] OpenPrivacy.org, An Annotated OpenPrivacy Bibliography
http://www.openprivacy.org/bibliography.shtml

Jonathan Schull received a Ph.D. in Biological Psychology from the University of Pennsylvania in 1980, taught at Haverford College until 1992, and then gave up tenure to found SoftLock Services, an early rights management company, that later changed its name to DigitalGoods.com. In 1999, Schull drafted the Open Ebook Forum's "Framework for the Epublishing Ecology. By 2000, when Stephen King published his e-book "Riding the Bullet", DigitalGoods was a 75 person company listed on the Nasdaq stock exchange. In 2001, the company closed its doors and liquidated its assets, including patents Schull authored in 1992. Schull now teaches Human Computer Interaction in the Interactive Media group, in the Information Technology Department of the Rochester Institute of Technology, and continues to pursue his long-standing interest in the dynamics of intelligent and adaptive networks.

Credits:

ODRL 2005

The Second International ODRL Workshop 2005

7-8 July 2005, ADETTI/ISCTE, Lisbon - Portugal

Editor-in-Chief

Carlos Serrão ADETTI / ISCTE

Editors

Renato Iannella, National ICT Australia (NICTA)
Susanne Guth, O_2 (Germany) GmbH
Carlos Serrão, ADETTI / ISCTE

Editorial Production

Ana Rita Leitão, ADETTI
Frederico Figueiredo, INESC

ADETTI - Associação para o Desenvolvimento das Telecomunicações e Técnicas de Informática
Avenida das Forças Armadas, Edifício ISCTE
1600-082 Lisboa, PORTUGAL
Tel: +351 21 7826480 Fax: +351 21 7826488

The ODRL Initiative
http://odrl.net/

Online Proceedings and Presentations: http://odrl.net/workshop2005/

Logo Credits

ODRL Initiative
Luis Taklim, Anyforms
Ana Rita Leitão

Sponsors

July 2005
ISBN: 1-740-64501-4