Interoperability between ODRL and MPEG-21 REL

Josep Polo, Jose Prados, Jaime Delgado
Universitat Pompeu Fabra (UPF), Departament de Tecnologia,
Pg. Circumval·lació 8, E-08003 Barcelona, Spain
{josep.polo, josep.prados, jaime.delgado}@upf.edu

Abstract

Two main Rights Expression Languages (RELs) exist to describe licenses governing the access to digital content: ODRL (Open Digital Rights Language) and MPEG-21 REL. Both RELs are powerful and complex enough. The use of different RELs could divide the network commerce in two separate factions. In this paper we propose a way for interoperability between them. They have many similarities that permit to translate expressions from one language into the other one. In the Distributed Multimedia Applications Group (DMAG) [12] we are developing utilities that permit to translate licenses between both RELs. Furthermore, the DMAG has developed a set of applications to generate and check licenses in both RELs.

This paper first describes the current situation of the RELs. Then the MPEG-21 REL and ODRL are introduced. Later, the interoperability between ODRL and MPEG-21 REL is exposed and the DMAG licenses generator and checker are described[1].

1 Rights Expression Languages

At present, network commerce of multimedia content is based on the trade of rights, but one of the limitations of Digital Rights Management (DRM) technology is due to the deficiency of means to express in an unambiguous, precise, machine-readable way the complex permissions on content. Furthermore it is not possible to create business relationships with distributors based on machine-readable licenses that can be automated to a significant degree. Consequently, today's business models have limited attraction to consumers and providers.

It is necessary to have a process by which the rights can be expressed in machine-readable licenses, guaranteed to be unambiguous and secure. A Rights Expression Language (REL) is the key to technical interoperability between proprietary DRM systems.

The basic component of a REL is the rights expression, which describes a permission granted to a user of protected content. Those rights expressions can be generated by any party authorised to grant permissions on content. In order for a rights expression language to be machine-readable, it must be based on a syntax that is recognised. Furthermore, there must be some measures built in, such as the ability to digitally sign rights expressions, so that their authenticity and tamper-resistance can be verified.

Several RELs have been proposed to describe licenses governing the terms and conditions of content access. In this field, ODRL [4, 5, 6] and MPEG-21 REL [16, 18] cover a prominent role. Both languages are powerful yet complex. This paper doesn't propose a way of analysing both languages to decide which of them is better, but it proposes to analyse the similarities and the interoperability of both languages.

[1] This work has been partly supported by the Spanish Ministry of Science and Technology (TIC2002-01336)

2 MPEG-21

At present, there is a standardisation committee, the Moving Picture Experts Group (MPEG), formally Working Group 11 of the ISO/IEC Joint Technical Committee, Sub-committee 29 [16], that covers most multimedia content subjects.

One of the standards produced by the MPEG is MPEG-21 [1]. Its aim is to offer interoperability in multimedia consumption and commerce. The standard is currently (after the March 2004 MPEG meeting) divided into 16 parts, most of them still under development. The number of parts may still increase.

Three of these parts are directly dealing with Digital Rights Management (DRM):

- **Part 4. Intellectual Property Management and Protection** (IPMP): provides the means to reliably manage and protect content across networks and devices.
- **Part 5. *Rights Expression Language*** (REL): specifies a machine-readable language that can declare rights and permissions using the terms as defined in the Rights Data Dictionary.
- **Part 6. Rights Data Dictionary** (RDD): specifies a dictionary of key terms required to describe users' rights.

This paper is focused in part 5: Rights Expression Language. This part explains the basic concepts of a machine interpretable language for expressing the rights and permissions of users that act on digital items, components, fragments, and containers.

2.1 MPEG-21 REL

The REL from MPEG-21 is based on the XrML proposal [10]. 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.

The core of MPEG-21 REL is the following four elements: principal, resource, right and condition (shown in Figure 1):

- **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. Example rights include play, print, issue, obtain, etc.
- **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. For example, a video file that a principal may play.
- **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.

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

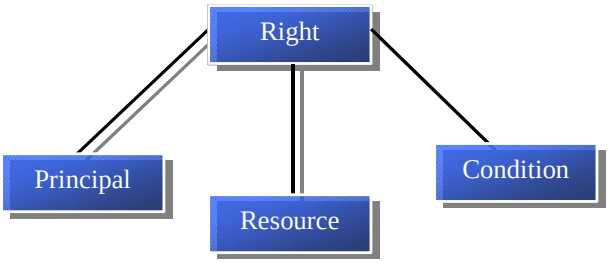

Figure 1. Core elements of MPEG-21 REL.

The MPEG-21 REL function is to express rights granted by some principals for specific resources and the conditions under which those rights apply. It does not provide any encryption functionality for content, though it does link to processes for ensuring the rights expressions themselves are tamper proof and capable of authentication.

The basic MPEG-21 REL element is the license. A license can contain one o more grants, the license issuer, that gives the grants that the license contains, and additional administrative information. Each grant must contain information to identify the four elements (principal, resource, right and condition) associated to it. Figure 2 shows a simple license structure.

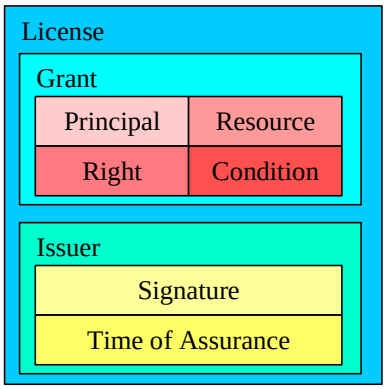

Figure 2. MPEG-21 REL license.

The license issuer who issues a license can digitally sign it, meaning that the issuer does really give the grants contained in it. Multiple issuers may sign a given license.

A grant is the part of an MPEG-21 REL license that conveys to an identified party the right to use a resource subject to certain conditions.

For example, consider an e-book named “Why Cats Sleep and We don’t” distributed to a consumer (Alice) that she can print 3 times. The MPEG-21 REL document 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 3 shows this example grant.

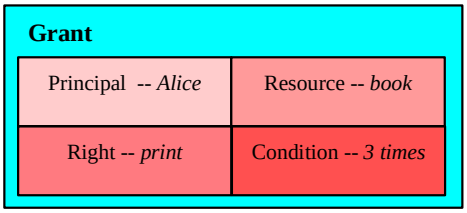

Figure 3. MPEG-21 REL grant example.

3 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, digital images, audio and movies, learning objects, computer software and other creations in digital form. This is an XML-based usage grammar.

ODRL is focused on the semantics of expressing rights languages and definitions of elements in the data dictionary. ODRL can be used within trusted or untrusted systems for both digital and physical assets (resources).

ODRL is based on an extensible model for rights expressions, which involves three core entities and their relationships. These are shown in Figure 4. They are:

- **Party** includes end users and Rights Holders. Parties can be humans, organisations, and defined roles. In the previous example, Alice is the party.
- **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. In the previous example, print is the right that includes the constrain of “3 times”.
- **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. In the previous example, the book is the asset.

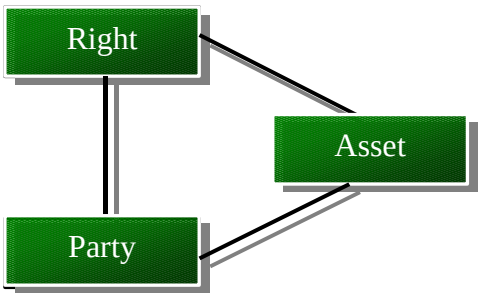

Figure 4. Core elements of ODRL.

4 Interoperability between ODRL and MPEG-21 REL

ODRL and MPEG-21 REL have many similarities: syntactically they are both based on XML, while structurally they both conform to the Stefik's axiomatic principles of rights modelling [2, 14].

A difference between ODRL and MPEG-21 REL is that ODRL seems more adapted to actual transactions in the commerce environment, whereas MPEG-21 REL has designs on broader cross-vertical applicability. ODRL's primitives map more directly onto the types of terms that are found in real-world commerce.

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

They have different entities, but these try to represent the same information. After analysing both languages, we can conclude that there are four main entities in a license:

- **Subject**: actor who performs some operation. In ODRL, it is the party and in MPEG-21 REL it is the principal.
- **Right**: what a subject can do to an object. In ODRL it is the permission (right) and in MPEG-21 REL it is represented by the right.
- **Object**: content acted upon by a subject. In ODRL it is the asset and in MPEG-21 REL it is the resource.
- **Condition**: describes when a right can be performed. In ODRL it is the constraint and is included in the permission (right), and in MPEG-21 REL it is the condition.

As an illustration, we can consider the previous example: Alice has got a license to print an e-book 3 times.

Intuitively, the *subject* of this example is “Alice”, the *object* is “book”, the *right* is “print” and the *condition* is “3 times”.

Figure 5 shows the ODRL license for this example.

```
<?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"
             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:asset>
      <o-ex:context>
         <o-dd:uid>urn:ebook.world/999999/ebook/rossi-000001</o-dd:uid>
         <o-dd:name>Why Cats Sleep and We don't</o-dd:name>
      </o-ex:context>
   </o-ex:asset>
   <o-ex:permission>
      <o-dd:print>
         <o-ex:constraint>
            <o-dd:count>3</o-dd:count>
         </o-ex:constraint>
      </o-dd:print>
   </o-ex:permission>

   <o-ex:party>
      <o-ex:context>
         <o-dd:name>Alice</o-dd:name>
      </o-ex:context>
   </o-ex:party>
</o-ex:rights>
```

Figure 5. ODRL example license.

Figure 6 shows the equivalent MPEG-21 REL license.

```
<?xml version="1.0" encoding="UTF-8" ?>
<r:license xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
      xmlns:mx="urn:mpeg:mpeg21:2003:01-REL-MX-NS"
      xmlns:r="urn:mpeg:mpeg21:2003:01-REL-R-NS"
      xmlns:sx="urn:mpeg:mpeg21:2003:01-REL-SX-NS"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="urn:mpeg:mpeg21:2003:01-REL-R-NS ../schemas/rel-r.xsd
           urn:mpeg:mpeg21:2003:01-REL-SX-NS ../schemas/rel-sx.xsd
           urn:mpeg:mpeg21:2003:01-REL-MX-NS ../schemas/rel-mx.xsd">
   <r:grant>
      <r:keyHolder>
         <r:info>
            <dsig:KeyName>Alice</dsig:KeyName>
         </r:info>
      </r:keyHolder>
      <mx:print />
      <r:digitalResource>
         <r:nonSecureIndirect URI="urn:ebook.world/999999/ebook/rossi-000001" />
      </r:digitalResource>
      <r:allConditions>
         <sx:exerciseLimit>
            <sx:count>3</sx:count>
         </sx:exerciseLimit>
      </r:allConditions>
   </r:grant>
</r:license>
```

Figure 6. MPEG-21 REL example license.

Table 1 shows the four main entities and their relationship with ODRL and MPEG-21 REL.

Table 1. The four main entities in the licenses.

ENTITY	ODRL	MPEG-21 REL
Subject	o-ex:party	r:keyHolder
Object	o-ex:asset	r:digitalResource
Right	o-ex:permission	r:grant
Condition	o-ex:constraint	r :allConditions

If we consider the similarities that can be seen in the previous example, as well as in the previous part of the paper, it can be concluded that the interoperability between both languages is possible. To transform an ODRL license into an MPEG-21 REL license, or vice versa, it is equivalent to transform a XML document to another XML document, where the information to represent is the same one, but with a different XML structure.

These are preliminary results. As it can be seen from the example, these licenses in ODRL and MPEG-21 REL are very simple. It is supposed that more complicated licensed will be more difficult to transform between ODRL and MPEG-21 REL, but it is also to be expected that the transformation will be done properly due to the languages similarity.

In order to obtain this transformation, XSL (Extensible Stylesheet Language) can be used. The XSL is one of the most important intricate specifications in the XML family. Using XSLT (XSL Transformation) [3] is not the only way to transform XML documents. A general purpose programming language like C, C++, or Java can also be used. XSLT has the advantage of being more lightweight than those languages and it is oriented to XML interaction. It is adequate for transformation and is well-equipped as a language to perform this main design goal. It allows to write programs that are much smaller than with a general purpose programming language. XSL can be broken in two parts: the said XSLT and XSL-FO (XSL Formatting Objects). 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.

In order to implement this interoperability, we have developed two utilities: one to transform an ODRL License into a MPEG-21 REL License and the other one to transform into the reverse direction. These utilities have been developed using XSL pages. The structures of both tools are very similar, but many differences exist at rules implementation level, given the syntactic and semantic differences of both languages. Figure 7 shows the XSLT document that allows to transform the previous ODRL license (Figure 5) to a MPEG-21 REL license (Figure 6). The other XSLT document (MPEG-21 REL to ODRL) is quite similar, so it is not shown.

```
<?xml version="1.0" encoding="UTF-8" ?>

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:o-
ex="http://odrl.net/1.1/ODRL-EX" xmlns:o-dd="http://odrl.net/1.1/ODRL-DD"
xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
      xmlns:mx="urn:mpeg:mpeg21:2003:01-REL-MX-NS"
      xmlns:r="urn:mpeg:mpeg21:2003:01-REL-R-NS"
      xmlns:sx="urn:mpeg:mpeg21:2003:01-REL-SX-NS"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="urn:mpeg:mpeg21:2003:01-REL-R-NS ../schemas/rel-r.xsd urn:mpeg:mpeg21:2003:01-
REL-SX-NS ../schemas/rel-sx.xsd urn:mpeg:mpeg21:2003:01-REL-MX-NS ../schemas/rel-mx.xsd">

<!-- Output file is a XML file -->
   <xsl:output method="xml"/>

<!-- Substitute root node from ODRL file to this MPEG-21 REL template -->

   <xsl:template match="/">
      <r:license xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
      xmlns:mx="urn:mpeg:mpeg21:2003:01-REL-MX-NS"
      xmlns:r="urn:mpeg:mpeg21:2003:01-REL-R-NS"
      xmlns:sx="urn:mpeg:mpeg21:2003:01-REL-SX-NS"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="urn:mpeg:mpeg21:2003:01-REL-R-NS ../schemas/rel-r.xsd
urn:mpeg:mpeg21:2003:01-REL-SX-NS ../schemas/rel-sx.xsd urn:mpeg:mpeg21:2003:01-REL-MX-NS
../schemas/rel-mx.xsd">
         <r:grant>
            <!-- Inside the grant, apply the rest of templates or replacements -->
            <xsl:apply-templates/>
         </r:grant>
      </r:license>
   </xsl:template>

<!— ODRL file "o-ex:party" node replacement for this MPEG-21 REL template -->
   <xsl:template match="o-ex:party">
      <r:keyHolder>
         <r:info>
            <dsig:KeyName><xsl:value-of select="o-ex:context/o-dd:name"/></dsig:KeyName>
         </r:info>
      </r:keyHolder>
   </xsl:template>

<!-- ODRL file "o-ex:asset" node replacement for this MPEG-21 REL template -->
   <xsl:template match="o-ex:asset">
      <r:digitalResource>
         <xsl:element name="r:nonSecureIndirect">
            <xsl:attribute name="URI"><xsl:value-of select="o-ex:context/o-dd:uid"/></xsl:attribute>
         </xsl:element>
      </r:digitalResource>
   </xsl:template>

<!-- ODRL file "o-ex:permission" node replacement for this MPEG-21 REL template -->
   <xsl:template match="o-ex:permission">
      <r:allConditions>
         <sx:exerciseLimit>
            <sx:count><xsl:value-of select="o-dd:print/o-ex:constraint/o-dd:count"/></sx:count>
         </sx:exerciseLimit>
      </r:allConditions>
   </xsl:template>

</xsl:stylesheet>
```

Figure 7. XSLT document that transforms the previous ODRL license in Figure 5, into the MPEG-21 REL license in Figure 6.

XSLT documents form a special class of XML documents. As all XML documents, it has one document element. This is the xsl:stylesheet element.

To specify the output method, the document has an xsl:output element. It is xml in this case.

The xsl:template element is the most important element in XSLT. It is the basis for matching patterns to perform transformation. The match attribute is used to match a pattern. The document has an xsl:template match="/" element. This element matches the root of a document. The transformation process always starts from an element that matches the root. This element allow us to substitute root node from the ODRL document to the MPEG-21 REL document.

The rest of nodes are inside the root node. This is the way to declare and call subroutines. Using an xsl:apply-templates element, we apply the rest of templates or replacements inside this root node to the other nodes.

The document has three more xsl:template elements: to transform o-ex:party to r:keyHolder (from ODRL to MPEG-21 REL), to transform o-ex:asset to r:digitalResource and to transform o-ex:permission to r:allConditions.

The xsl:element is used to construct an element in the result tree. The mandatory attribute is the name attribute, which specifies the name of the element to be generated.

The xsl:attribute element is used inside the template that match o-ex:asset, and it is used to add an attribute in the result tree.

The xsl:value-of element returns the string value of the expression given in the select attribute.

5 REL Tools

We have also developed a set of applications that permit to generate licenses in the REL languages: ODRL and MPEG-21 REL [7, 8, 9, 13]. The internal structure of the applications has similarities and differences between the two language versions, but from the point of view of the formal structure they are equivalents. For this reason, only the ODRL applications are described below. The MPEG-21 REL applications have the same formal description.

5.1 DMAG ODRL License Creator

The DMAG ODRL License Creator (DOLC) is a software implementation that creates ODRL Licenses. This software has been developed in Java. It can run on MS-Windows and Linux platforms. At present, the DOLC can only create basic licenses. The license types are offer and agreement. It is expected that, shortly, the software will be able to generate more complex licenses.

This software can create XML documents representing valid ODRL Licenses. Their implementation is based on the Document Object Model (DOM) API for HTML and XML documents. The DOM API provides a structural representation of the document, and it defines the way that a structure is to be accessed from a script. Essentially, it connects web pages or XML documents to programming languages.

From the point of view of a user, the DOLC has been implemented as a web application. It is composed by a web page containing an HTML form, a servlet for processing the information

introduced in the form and to generate the ODRL License, and finally, a web page containing the ODRL License created.

In order to syntactically validate the license against their schemas, another application has been developed, the DMAG Schema Checker (DSC), which is described in 5.2. So, the DSC software has been used to verify the validity of the ODRL License created by the DOLC.

The information needed to generate the license is introduced by means of the HTML form. This module allows the introduction of: asset, rights, requirements, constraints, parties, etc.

This information is sent to the servlet, which is installed in a servlet engine. Tomcat [11] is the servlet engine used. The License Creator servlet, extracts the information introduced in the web page form by the user that is creating the license, checking that all the needed fields have been filled. If all necessary information has been introduced, the XML file containing the license is created.

The license is conformant with its schemes using the DSC. Finally, the result is shown to the user, who can see the generated license.

Next steps to be taken are the addition of more complex permissions and/or constraints and the study of more complex licenses. The tools used to implement the DOLC application allow these additions in an easy way.

Figure 8 shows the structure of the DOLC developed.

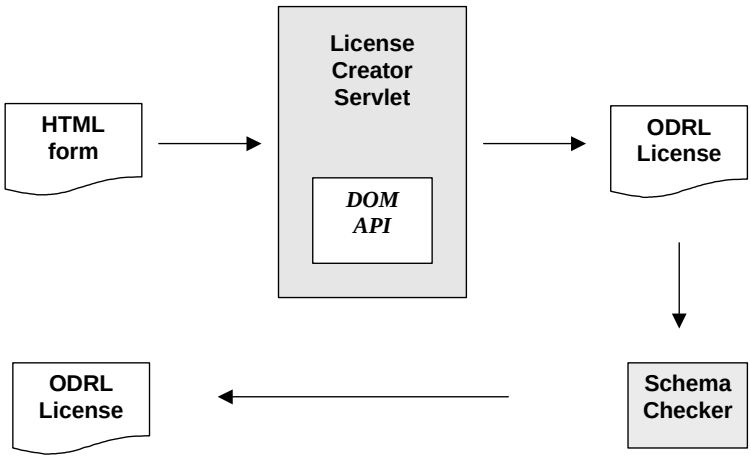

Figure 8. DMAG ODRL License Creator (DOCL) and its relation with the DMAG Schema Checker (DSC)

5.2 DMAG Schema Checker

As it was explained above, the DMAG Schema Checker (DSC) is an application that validates syntactically ODRL and other types of licenses such as MPEG-21 REL Licenses, against the XML schemas used by the licenses. 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 [19]. The software validates syntactically a document with an ODRL license specified by the user.

The output of the DSC is a message reporting if the license is syntactically valid or not, according to the XML Schemas specified within the license. If the license is not valid, the DSC will inform about the reasons why.

Figure 9 shows the structure of the DSC developed.

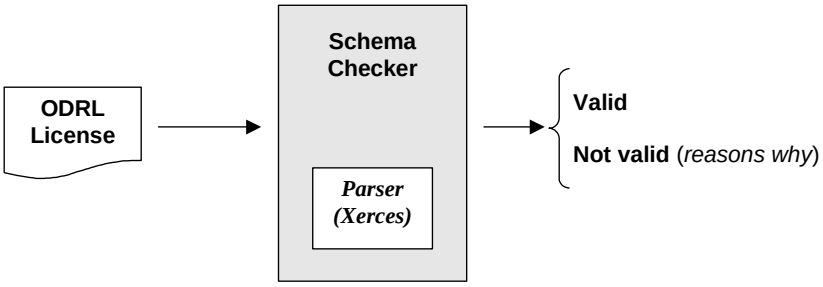

Figure 9. DMAG Schema Checker (DSC)

6 Conclusions and future work

A set of applications has been presented to permit interoperability between ODRL and MPEG-21 REL licenses. In the DMAG, we have developed utilities that permit to translate ODRL licenses into MPEG-21 REL licenses and in the other way around. The current licenses are simple, but we are working to expand the interoperability to more complex licenses.

Furthermore, a set of applications has been described to generate and check licenses in ODRL and MPEG-21 REL. At present, they can generate non-complex licenses, but we are also working to extend the capabilities of the applications.

The objective of our future work is to expand the scope of this set of applications (generators and converters) to permit that every system could work in ODRL or MPEG-21 REL without distinction, transparently to the user, generating the appropriate license, and, furthermore, when receiving a license, it would not matter the REL license format, because the system could work without distinction.

References

1. Burnett et al. *MPEG-21 Goals and Achievements.* IEEE Computer Society, October-November 2003

2. C.N. Chong et al. *Comparing Logic-based and XML-based Rights Expression Languages.* Confederated International Workshops: On The Move to Meaningful Internet Systems. Catania, Sicily, Italy. Published by Springer-Verlag. November, 2003.

3. K.Y. Fung. *XSLT Working with XML and HTML*. Addison-Wesley.

4. S. Guth et al. *Experiences with the Enforcement of Access Rights Extracted from ODRL-based Digital Contracts*. ACM Workshop on Digital Rights Management, Washington D.C./USA, October 2003.

5. S. Guth, E. Koeppen. *Electronic Rights Enforcement for Learning Media*. IEEE International Conference on Advanced Learning Technologies, Kazan/Russland, September, 2002.

6. R. Iannella. *Open Digital Rights Language (ODRL)* Version 1.1. http://odrl.net. August 2002.

7. S. Llorente, J. Delgado, E. Rodríguez. Enhanced versions of DMAG REL Reference Software. ISO/IEC JTC1/SC29/WG11 - MPEG2003/M10286. December 2003. http://dmag.upf.edu/DMAGRELTools/m10286.pdf.

8. S. Llorente, J. Delgado, E. Rodríguez. *DMAG REL License Creator*. ISO/IEC JTC1/SC29/WG11 - MPEG2003/M10040. October 2003. http://dmag.upf.edu/DMAGRELTools/m10040.pdf.

9. E. Rodríguez, J. Delgado, S. Llorente. *DMAG REL Schema Checker*. ISO/IEC JTC1/SC29/WG11 - MPEG2003/M10039. October 2003. http://dmag.upf.edu/DMAGRELTools/m10039.pdf.

10. X. Wang et al. *XrML – eXtensible rights Markup Language*. ACM Workshop on XML Security, Nov. 22, 2002, Fairfax VA, USA.

11. Apache Tomcat. http://jakarta.apache.org/tomcat/index.html.

12. Distributed Multimedia Applications Group (DMAG), Technology Dept., Universitat Pompeu Fabra. http://dmag.upf.edu

13. DMAG MPEG REL reference software. http://dmag.upf.edu/DMAGRELTools/Index.htm

14. eXtensible Rights Markup Language (XrML), Version 2.0. DRM watch. GiantSteps, Media Technology Strategies. http://www.giantstepsmts.com/DRM%20Watch/xrml20.htm

15. Information Technology - Multimedia Framework - Part 5: Rights Expression Language. FDIS 21000-5:2003(E). ISO/IEC JTC1/SC29/WG11/N5839.

16. Moving Picture Expert Group. ISO/IEC JTC/SC29 WG11. http://www.chiariglione.org/mpeg/index.htm

17. Moving Picture Expert Group. ISO/IEC JTC/SC29 WG11. Programme of Work. http://www.itscj.ipsj.or.jp/sc29/29w42911.htm#MPEG-21.

18. Rightscom Ltd. *The MPEG-21 Rights Expression Language. A white paper.* http://www.rightscom.com/files/MPEG21_RELwhite_paper.pdf

19. Xerces Parser. http://xml.apache.org/xerces-j/