IPTC Standards

RightsML

Version 1.1 Experimental Phase 2

Specification of a Profile and Vocabulary for the communication of usage rights in ODRL 2.0

Copyrights

Acknowledgments

About the Standard

Specification versioning history

Version	Approval Date	Approved by	Remarks
1.0EP1	2012-03-21	IPTC Standards Committee	Specification is at Experimental Phase level.
1.1EP2	2012-10-23	IPTC Standards Committee	Specification is at Experimental Phase level.

Document revision history

Version	Issue Date	Authored/revised by	Remarks
1	2013-09-24	Michel Steidl for the RightsML-Dev group	

About this Document

This document specifies the IPTC rights markup language standard RightsML which is a profile and vocabulary for the machine-readable communication of usage rights in ODRL 2.0.

Status of this Document

This document is under the governance of the Rights Expression Language Working Group of the IPTC.
This is a specification document which was endorsed by the IPTC members and may be updated, replaced or obsoleted by other documents at any time.

This Experimental Phase document is available in the directory http://iptc.org/std-dev/RightsML/1.1EP2/specification/ which can be accessed on the Web and any updates to this document will be published there.
Versions of the final standard will be published in a subdirectory of http://iptc.org/std/RightsML/ corresponding to the version number of the standard.

Table of Contents

1   Introduction

This document specifies a Profile of the Open Digital Rights Language (ODRL) Version 2.0 for application in the communication of usage policies, primarily in association with the licensed distribution and use of news content by news gathering agencies, news publishers, news licensing organisations, business intermediaries and business consumers in the online news market-place.

ODRL V2.0 defines a language for expression of rights and policies. This language has a number of key components:

- A Core Model, which defines the main structures and key concepts of the language.
- A Common Vocabulary, which defines a set of terms that may be used across all applications of ODRL V2.0.
- An XML encoding which defines the serialisation of ODRL 2.0 expressions in XML format. This is just one of a number of encodings and serializations that are under development; others include RDF and JSON.

The ODRL V2.0 Core Model describes a number of application scenarios, many of which are not applicable to the RightsML 1.1 use case. Not all the terms defined in the Common Vocabulary are applicable to this use case. This document therefore specifies a particular scenario for application of ODRL V2.0, and the vocabularies that may be used in that scenario.

References

In addition to the specifications of RightsML 1.1 the IPTC also maintains a developer site for RightsML implementers: see http://dev.iptc.org/RightsML. This site helps to implement the specifications by providing guidelines and best practices. Further it provides an open forum for discussing RightsML.

This RightsML specification builds on these ODRL 2.0 specification documents:
ODRL V2.0 – Core Model – Final Specification: 24 April 2012
(http://www.w3.org/community/odrl/two/model/)
ODRL V2.0 – Common Vocabulary – Final Specification: 24 April 2012
(http://www.w3.org/community/odrl/two/vocab/)
ODRL V2.0 – XML Encoding – Final Specification: 24 April 2012
(http://www.w3.org/community/odrl/two/xml/)

Any inclusions of ODRL V2.0 specifications are only made to specify how the RightsML 1.1 Profile and Vocabulary builds on this framework. The IPTC explicitly waives any claim on the copyright in the ODRL V 2.0 specifications.

RightsML use case

The RightsML use case is based upon the requirement, in the specific context of news syndication, to be able to associate a usage policy with an item of content for which usage rights are assigned by a licensing rights holder to a licensee. The assumption is that the licensee may not necessarily be the final consumer of the item in question, but is a licensed business entity that may wish to make any of a variety of permitted commercial uses of a content item, including using the item in the delivery of their own products or services to their own customers. The usage policy may therefore need to cover both the use that is made of the item by the immediate licensee to whom rights are being granted and the duty of the licensee to communicate specific usage policy terms to their own customers, associated with delivery of any content items or derivatives to their customers.

The requirement is to enable communication of the usage rights and constraints that are specific to a particular item. These might be delivered with the item – whether embedded in the item, or embedded in the communication payload that includes the item – or communicated separately.

RightsML 1.1 Experimental Phase and future development

The RightsML 1.1 Profile of ODRL 2.0 aims to satisfy this use case. Piloting activities have indicated that the Profile does, to a worthwhile extent, meet the requirements of this use case, insofar as the piloting activities have been able to test this. Further experimental implementations would be expected to highlight any gaps in the Profile, especially in the vocabulary.

It is anticipated that it will be necessary to extend the RightsML 1.1 Profile from time to time, especially the vocabulary. On the assumption that the market for digital services in news delivery will continue to evolve rapidly, new types of usage will arise for which usage policies cannot be adequately expressed without extending the Profile. It is anticipated that such extensions to the RightsML 1.1 Profile will, in the first instance, be private extensions, worked out between the business partners most immediately concerned. In the longer term it will clearly benefit the market as a whole for such extensions to be considered for incorporation into the RightsML 1.1 Profile.

Using "namespaces" to identify different vocabularies

ODRL 2.0 anticipates that applications will need to extend the Common Vocabulary in a variety of different ways. The RightsML 1.1 Profile recognises that terms from a variety of different vocabularies are needed to provide the rich forms of expression that many applications will require. At least three vocabularies are needed for any realistic application:

- The ODRL Core Model
- The ODRL Common Vocabulary
- The RightsML Vocabulary.

Other vocabularies may be needed in specific cases, including both proprietary and standard vocabularies.

The provenance of all vocabularies must be indicated by explicitly specifying the vocabulary from which each term is taken. Each vocabulary should be represented by a Uniform Resource Identifier (URI) that has been issued by the owner or provider of the vocabulary. This URI, known as the "namespace URI" for the vocabulary in question, need not necessarily resolve to a web resource, although it may do so and if so, the web resource should relate directly to the vocabulary in question.

The precise way in which a vocabulary term is expressed will depend upon the chosen encoding. In some encodings, such as XML, it may be appropriate or necessary to declare a "namespace prefix", a short prefix that can be attached to the term in order to specify the vocabulary to which it belongs. If a term is expressed using a namespace prefix, the prefix must be declared and must be associated with the correct namespace URI for the vocabulary in question, following the rules of the chosen encoding for declaring namespaces and prefixes.

The namespace URIs for the ODRL 2.0 Core Model, the ODRL 2.0 Common Vocabulary and the RightsML 1.1 Vocabulary are given below.

Vocabulary	Namespace URI
ODRL 2.0 Core Model	http://w3.org/ns/odrl/2/
ODRL 2.0 Common Vocabulary	http://w3.org/ns/odrl/vocab#
RightsML Vocabulary	http://iptc.org/std/RightsML/2011-10-07/

The RightsML 1.1 Profile of ODRL 2.0

The RightsML 1.1 Profile is specified in accordance with the guidance on specifying Profiles that is given in Section 4 of the ODRL 2.0 Core Model.

Additions to the Core Model

The RightsML 1.1 Profile does not add any features to the Core Model. The range of values of certain attributes is extended, as described below.

Implementation Guidelines of Core Model features

Policy entity

In implementations of the RightsML 1.1 Profile it is recommended that the value of the **type** attribute of a Policy should generally be **"set"**, because the Set Scenario does not require any Parties to be identified in the expression of a Permission or Prohibition.

Asset entity

The value of the **uid** attribute of an Asset identifies either a specific resource (content item) or a group of resources or implicitly only a resource:

- If a rights expression applies explicitly to a single asset only the uid attribute MUST provide the identifier for this asset.
- If a rights expression applies explicitly to a group of assets the uid attribute MUST provide an identifier which identifies the group. How this is achieved has to be defined and provided by the owner of the assets.
- If a rights expression implicitly applies to all assets which point to the ODRL policy document which includes this asset property MUST use a uid attribute with the character * (0u002A) as value. Example: <asset uid="*" relation="http://w3.org/ns/odrl/vocab#target" />

  In this case the asset properties of all Permissions or Prohibitions SHOULD include such a wildcard uid attribute.

Party entity
The value of the **uid** attribute of a Party may identify either a specific party or a category of parties. The range of values of the **uid** attribute is extended to allow a party category to be represented by a term from a specified vocabulary. It is recommended that URIs only be used to identify a specific party as the Party entity. Vocabulary terms must only be used to identify a Party entity to be a category.

Role entity
No change.

Permission entity
No change.

Duty entity
No change.

Prohibition entity
No change.

Action entity
No change.

Constraint entity
No change.

Experimental features in the ODRL 2.0 Core Model
The RightsML 1.1 Profile does not include any of the experimental features described in Section 5 of the ODRL 2.0 Core Model.

ODRL 2.0 Common Vocabulary
The namespace for all terms of this vocabulary is: http://w3.org/ns/odrl/vocab#

Some terms defined in the ODRL 2.0 Common Vocabulary are explicitly recommended for the use with the RightsML 1.1 Profile, the other terms are considered to be less useful in the RightsML 1.1 use case. In some cases alternative terms have been specified in the RightsML 1.1 Vocabulary.

Policy types
The preferred term for use in most implementations of the RightsML 1.1 Profile is **agreement,** other policy types may be used.

Actions
The following actions are **recommended** by the RightsML 1.1 Profile:
- **aggregate**
- **annotate**
- **attribute**
- **delete**

- **derive / modify**
- **display / present**
- **export / transform**
- **extract**
- **give**
- **include**
- **index**
- **inform**
- **nextPolicy**
- **obtainConsent**
- **pay**
- **play / present**
- **print**
- **share**
- **translate**

Other actions of the ODRL Action vocabulary may be used without violating the RightsML profile.

Some ODRL actions are redefined by the RightsML 1.1 Action Vocabulary – see The RightsML 1.1 Vocabulary -, a distinction is made by the different namespace identifiers.
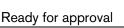
Constraints

Attribute 'name' vocabulary
No change of the ODRL specifications.

Attribute 'operator'
No change of the ODRL specifications.

Attribute 'rightOperand'
No change of the ODRL specifications.

Party and Role
No change of the ODRL specifications.

Asset and Relation
No change of the ODRL specifications.

The RightsML 1.1 Vocabulary

The namespace for all terms of this vocabulary is: http://iptc.org/std/RightsML/2011-10-07/

Note: in the XML-encoded examples the following namespace prefixes are used

Prefix	Vocabulary namespace
o:	ODRL 2.0 Core Model, see Using "namespaces" to identify different vocabularies
ov:	ODRL 2.0 Common Vocabulary, see Using "namespaces" to identify different vocabularies
rml:	RightsML Vocabulary, see Using "namespaces" to identify different vocabularies
p:	Vocabulary created and maintained by the assigner with its proprietary namespace

Assets

No terms associated with Asset entities are currently defined.

Parties

No terms associated with Party entities are currently defined.

Actions

The following Action terms are defined.

Identifier	*Term*	*Definition*
archive	Archive	Preserve a persistent copy of a content item so that it can be retrieved over a significant period of time, which may be bounded (i.e. have an end-date or end-time) or unbounded. This Action term redefines the term ov:archive from the ODRL Vocabulary.
copy	Copy	Make an exact copy of a content item. Usually preparatory to another action such as in copying to create a backup archive, or in copying to create a modified version. On its own does not necessarily imply that the copy has to be persistent – it may be transient. This Action term redefines the term ov:copy from the ODRL Vocabulary.
distribute	Distribute	The act of distributing, displaying and/or performing an Asset to licensed recipients. Not "publicly" distribute, as in the ODRL 2.0 Common Vocabulary. This Action term redefines the term ov:distribute from the ODRL Vocabulary.
removeFromService	Remove from product or service	The act of removing the specified Asset from the Assignee's product or service. Typically expressing a Duty to be performed on request, in association with permission to perform the action **addToService**.
use	Use	When associated with a permission, the act of making any reasonable use of the Asset, usually constrained to be for a specific purpose. When associated with a prohibition, the act of making any use whatsoever of the Asset.

Constraints

Attribute 'name'

The following terms are defined for use as names of Constraint entities.

Identifier	Term	Definition
actionRequestReceived	Action request received	The state that a request to perform an Action has been received. The right operand must contain either **true** or **false**. The constraint must be used in a Duty.
Example of use: see **requestedActionsPerformed**.		

Example of use of recipient: It is permitted to distribute the Asset on condition that the recipient is not party X.

```
<o:permission>
    <o:asset uid="targetAssetURI" relation="ov:target"/>
    <o:action name="rml:distribute"/>
    <o:constraint name="rml:recipient" operator="ov:neq" rightOperand="p:X"/>
</o:permission>
```

Identifier	Term	Definition
requestedActionsPerformed	Requested actions performed	The state that a requested Action has been performed, e.g. removal. The right operand must contain either true or false. The constraint must be used in a Duty, and there must be at least one Duty to perform an Action on request within the same Permission.

Example of use: It is permitted for the Assignee to add the Asset to their service, but this entails a Duty to remove the Asset from the service in the event that a removal request is received, and to inform the Assigner when the Asset has been removed.

```
<o:permission>
    <o:asset uid="targetAssetURI" relation="ov:target"/>
    <o:action name="rml:addToService" />
    <o:duty>
        <o:asset uid="targetAssetURI"/>
        <o:action name="rml:removeFromService"/>
        <o:constraint name="rml:actionRequestReceived" operator="ov:eq"
            rightOperand="p:true"/>
    </o:duty>
    <o:duty>
        <o:action name="ov:inform"/>
        <o:constraint name="rml:requestedActionsPerformed" operator="ov:eq"
            rightOperand="p:true"/>
        <o:party function="ov:informedParty" uid="licensorURI"/>
    </o:duty>
</o:permission>
```

Attribute 'rightOperand'

The following terms are to be used as category identifiers in the rightOperand attribute:

Identifier	Term	Definition
requestReceivedDateTime	Request received date-time	A date-time category: the date and optionally the time at which the request to perform an Action was received. To be used in a **dateTime** Constraint on a Duty to perform a requested Action, when specifying the date and optionally the time at which the requested Action must be performed relative to when the request was

Identifier	Term	Definition
		received.

Example of use: It is permitted for the Assignee to add the Asset to their service, but this entails a Duty to remove the Asset from the service in the event that a removal request is received, and to inform the Assigner within 24 hours that the Asset has been removed.

```
<o:permission>
    <o:asset uid="targetAssetURI" relation="ov:target"/>
    <o:action name="rml:addToService"/>
    <o:duty>
        <o:asset uid="targetAssetURI"/>
        <o:action name="rml:removeFromService"/>
        <o:constraint name="rml:actionRequestReceived" operator="ov:eq"
            rightOperand="p:true"/>
    </o:duty>
    <o:duty>
        <o:action name="ov:inform"/>
        <o:constraint name="rml:requestedActionsPerformed" operator="ov:eq"
            rightOperand="p:true"/>
        <o:constraint name="ov:dateTime" operator="ov:ltEq"
            rightOperand="rml:requestReceivedDateTime + PT24H"/>
        <o:party function="ov:informedParty" uid="licensorURI"/>
    </o:duty>
</o:permission>
```

Identifier	Term	Definition
serviceDevelopment **serviceDemonstration** **serviceTesting**	Service development Service demonstration Service testing	To be used in a **purpose** Constraint

Example of use: The assignee is permitted to make a copy of the Asset for service testing purposes.
```
<o:permission>
    <o:asset uid="targetAssetURI" relation="ov:target"/>
    <o:action name="ov:copy"/>
    <o:constraint name="ov:purpose" operator="ov:eq"
        rightOperand="rml:serviceTesting"/>
</o:permission>
```

=== END of document ===