Open Digital Rights Language (ODRL)

Version: 0.9
Date: 2001-06-29

URI: <http://odrl.net/0.9/ODRL-09.pdf>
URI: <http://odrl.net/0.9/ODRL-09.html>

Editor: Renato Iannella <renato@iprsystems.com>

Status

This document is a **work-in-progress** and may be updated and/or replaced by other documents at any time.

The intention is to promote this draft document amongst multiple communities interested in the expression of Digital Rights Management statements and semantic interoperability across these communities.

ODRL will be standardised via an appropriate, open, and non-competitive organisation with an open process for the future maintenance of the standard.

ODRL has no license requirements and is available in the spirit of “open source” software.

Comments are welcome to the editors from all interested parties.

Change Bars (in the PDF document only) indicate modifications from the previous version.

Table of Contents

1 Overview

Digital Rights Management (DRM) involves the description, layering, analysis, valuation, trading and monitoring of the rights over an enterprise's assets; both in physical and digital form; and of tangible and intangible value. DRM covers the digital management of rights - be they rights in a physical manifestation of a work (eg a book), or be they rights in a digital manifestation of a work (eg an ebook). Current methods of managing, trading and protecting such assets are inefficient, proprietary, or else often require the information to be wrapped or embedded in a physical format.

A key feature of managing online rights will be the substantial increase in re-use of digital material on the Web as well as the increased efficiency for physical material. The pervasive Internet is changing the nature of distribution of digital media from a passive one way flow (from Publisher to the End User) to a much more interactive cycle where creations are re-used, combined and extended ad infinitum. At all stages, the Rights need to be managed and honoured with trusted services.

Current Rights management technologies include languages for describing the terms and conditions, tracking asset usages by enforcing controlled environments or encoded asset manifestations, and closed architectures for the overall management of rights.

The Open Digital Rights Language (***ODRL***) provides the semantics for DRM in open and trusted environments whilst being agnostic to mechanisms to achieve the secure architectures.

1.1 The Bigger Picture

It is envisaged that ***ODRL*** will "plug into" an open framework that enables peer-to-peer interoperability for DRM services. However, ***ODRL*** can also be used as an mechanism to express rights statements on its own and to plug into existing DRM architectures/ frameworks.

Digital Rights Management (DRM) has traditionally been flavoured with security and encryption as a means to solve these issues, that is, Digital Rights Enforcement. This is the first-generation of DRM and represents a substantial narrowing of its real and broader capabilities. Second-generation DRM systems will provide these additional functions and support end-to-end supply chain DRM services.

DRM systems should be based on and designed around open functional architectures and a robust and extensible information model. See [IANNELLA] for an overview of DRM Architectures and [ERICKSON] for DRM-enabled digital object models. The layers and relationships of rights can also quickly become very complex [HIGGS] and DRM systems must be designed to cater for this intricacy.

1.2 About this Specification

This document, along with its normative references, includes all the specification necessary for the implementation of interoperable ***ODRL*** applications.

The key words *must, must not, required, shall, shall not, should, should not, recommended, may,* and *optional* in this specification are to be interpreted as described in [RFC2119] which defines the significance of each particular requirement.

Examples used in this document are for demonstration purposes only.

1.3 Scope

ODRL complements existing analogue rights management standards by providing digital equivalents, and supports an expandible range of new services that can be afforded by the digital nature of the assets in the Web environment. In the physical environment, ***ODRL*** can be used to enable machine-based processing for Rights management.

ODRL is a standard vocabulary for the expression of terms and conditions over assets. ***ODRL*** covers a core set of semantics for these purposes including the rights holders and the expression of permissible usages for asset manifestations. Rights can be specified for a specific asset manifestation (format) or could be applied to a range of manifestations of the asset.

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. However, ***ODRL*** does not determine the capabilities nor requirements of any trusted services (eg for content protection, digital / physical delivery, and payment negotiation) that utilises its language. Clearly, however, ***ODRL*** will benefit transactions over digital assets as these can be captured and managed as a single rights transaction. In the physical world, ***ODRL*** expressions would need an accompanying system with the distribution of the physical asset.

ODRL defines a core set of semantics. Additional semantics can be layered on top of ***ODRL*** for third-party value added services.

ODRL does not enforce or mandate any policies for DRM, but provides the mechanisms to express such policies. Communities or organisations, that establish such policies based on ***ODRL***, do so based on their specific business or public access requirements.

ODRL depends on the use of unique identification of assets. This is a very difficult problem to address and to have agreement across many sectors and is why identification mechanisms and policies of the assets is outside the scope of ***ODRL***. Sector-specific versions of ***ODRL*** may address the need to infer information about the asset manifestation from its unique identifier.

The ***ODRL*** model is based on an analysis and survey of sector specific requirements (including models and semantics), and as such, aims to be compatible with a broad community base. ***ODRL*** intends to meet the common requirements for many sectors and has been influenced by the ongoing work and specifications / models of the following groups:

- The <indecs> Project [INDECS]
- Electronic Book Exchange Working Group [EBX]
- International Federation of Library Associations [IFLA]
- DOI Foundation [DOI]
- ONIX International [ONIX]
- Moving Pictures Expert Group [MPEG]
- IMS Global Learning Consortium [IMS]
- Dublin Core Metadata Initiative [DCMI]
- Propagate Project [PROPAGATE]
- OpenEBook Forum [OEBF]
- Publishers Requirements for Metadata [PRISM]
- Association of American Publishers [AAP]
- Digital Imaging [DIG35]

ODRL proposes to be compatible with the above groups by defining an independent and extensible set of semantics. ***ODRL*** does not depend on any media types as it is aimed for cross-sector interoperability.

1.4 Specification Overview

The ***ODRL*** specification contains three main sections:

- Section 2 describes the model for the ***ODRL*** expression language.
- Section 3 describes the semantics of the ***ODRL*** data dictionary elements.
- Section 4 describes the XML syntax used to encode the ***ODRL*** expressions and elements.

The formal Expression Language and Data Dictionary elements are defined in:

- Appendix A: ODRL Expression Language XML Schema (Normative)
- Appendix B: ODRL Data Dictionary XML Schema (Normative)

The next release will include details on:

- Extending ***ODRL*** (Section 5 "ODRL Extensibility")
- Signing ***ODRL*** expressions (Section 6 "Signing ODRL Expressions")
- Transporting ***ODRL*** expressions (Section 7 "Transporting ODRL Expressions")

2 ODRL Expression Language

The models for the ***ODRL*** language contain the structure and core semantics for the expressions. These models provide the overall framework for the expressions into which elements (from DRM data dictionaries) can be applied.

Note: The figures in this section cover both the Expression Language entities (shown as shadowed rectangles with "EX" in the top left corner) and the Data Dictionary elements (shown as rectangles with "DD" in the top left corner).

2.1 ODRL Foundation Model

ODRL is based on an extensible model for rights expressions which involves a number of core entities that are related to each other. This ***ODRL*** Foundation Model is shown in Figure 1.

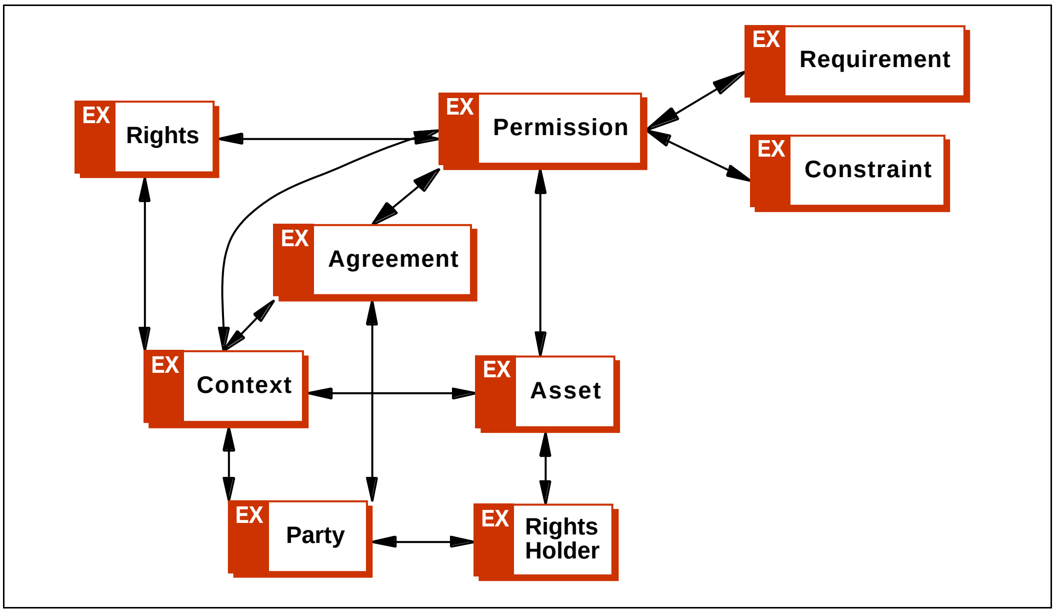

Figure 1. ODRL Foundation Model

The top-level Rights entity consists of an aggregation of the following entities:

- Permissions (see Section 2.2 "ODRL Permission Model")
- Constraints (see Section 2.3 "ODRL Constraint Model")
- Requirements (see Section 2.4 "ODRL Requirement Model")
- Rights Holders (see Section 2.5 "ODRL Rights Holder Model")
- Contexts (see Section 2.6 "ODRL Context Model")
- Agreements (see Section 2.7 "ODRL Agreement Model")
- Parties
- Assets

A Permission is the allowed activities over an Asset. Permissions may also be bound by Constraints and oblige Requirements from parties. Assets have Rights Holders (which are identifiable parties holding intellectual property rights over the asset).

Typically, these expressions allow for generic Rights to be declared over assets. Additionally, ***ODRL*** can support expressions when individual parties reach an Agreement (ie a specific instance of an expression) between a set of Permissions over Assets and Parties.

Most entities in the model can support a specific Context. A Context, which is relative to the entity, can describe further information about that entity or the relationship between entities.

For example, the Context of an Agreement may specify the date of the transaction, the Context of a Party may specify their role.

The general description of the Party and Asset entities is outside the scope of ***ODRL***. What is in scope is that these entities must be referenced by using unique identification mechanisms (such as [URI], [DOI], [ISBN] etc). Both of these entities contain a mechanism to refer to their unique identifiers, as well as a Context.

The Asset entity (sometimes referred to as a Work, Content, Creation, or Intellectual Property), is viewed as a whole entity. If the Rights are assigned at the Asset's subpart level, then such parts would require to also be uniquely identifiable. However, ***ODRL*** can specify constraints on subparts of the asset. Additionally, Assets can be identified as to their layer of intellectual property as defined by the IFLA model [IFLA]. These include Work, Expression, Manifestation, and Item. This features allows rights to be expressed over non-tangible assets.

These eight core Entities together allow for a wide and flexible range of ***ODRL*** expressions to be declared. The entities (except for Asset and Party) are further explained in the following sections, including XML pseudo-examples.

2.1.1 Foundation XML Example

The ***ODRL*** Foundation Model can be expressed using XML. A pseudo-example is shown Example 1.

```
<rights>
    <context> ... </context>
    <asset> ... </asset>
    <permission>
        <permission-type>
            <requirement> ... </requirement>
            <constraint> ... </constraint>
        </permission-type>
    </permission>
    <rightsholder>
        <party>
            <context> ... </context>
        </party>
    </rightsholder>
    <agreement>
        <party> ... </party>
        <permission> ... </permission>
    </agreement>
</rights>
```

Example 1.Foundation Model XML Syntax

Complete and formal syntactical examples are given in Section 4 "ODRL XML Syntax".

2.2 ODRL Permission Model

ODRL supports the expression of Permissions. This is the recognised set of activities allowed over the Asset. The ***ODRL*** Permission Model is shown in Figure 2.

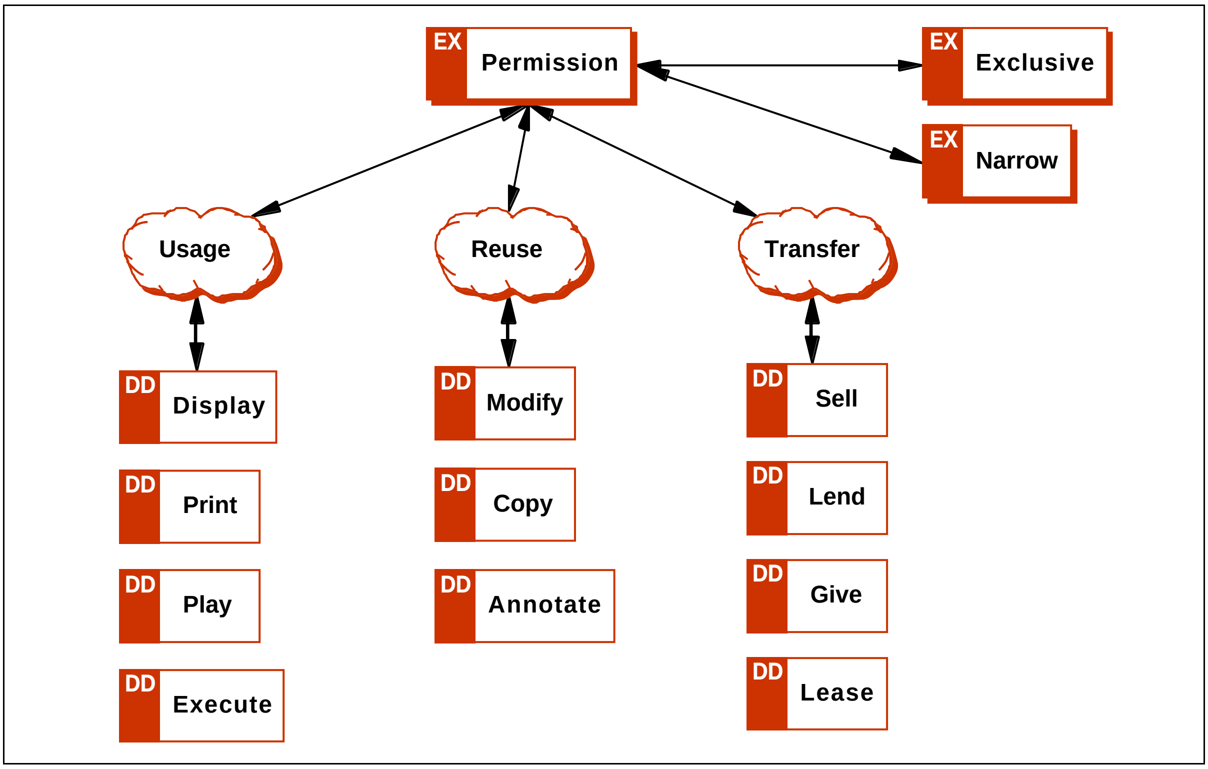

Figure 2. ODRL Permission Model

The Permission entity consists of an aggregation of three abstract entities. The abstract entities (depicted as clouds in Figure 2) are solely used to group similar permissions, and include:

- Usage - indicates a set of methods in which the Asset can be consumed (realised with: Display, Print, Play, Execute).
- Reuse - indicates a set of operations in which the Asset (or portions of it) can be re-utilised (realised with: Modify, Copy, Annotate).
- Transfer - indicates a set of procedures in which the rights over the Asset can be transferred (realised with: Sell, Lend, Give, Lease).

Note: The listed elements above (eg Display, Print etc) are defined in full in Section 3 "ODRL Data Dictionary Semantics" and form the basis of the ***ODRL*** Data Dictionary.

Additionally, the Permissions support:

- an “Exclusivity” attribute that indicates if the granted permission is confined to a nominated Party.
- a “Narrow” attribute that indicates if the granted permission can be re-issued (down-stream) with a less restricted scope.

A Permission must be associated with one or more Assets. A Permission can be associated with zero or more Constraints.

Important Note:

A Permission that is not specified in any Rights Expressions is not granted. That is, no assumptions should be made in regard to Permissions if they are not explicitly mentioned in the ***ODRL*** expression.

2.2.1 Permission XML Example

The ***ODRL*** Permission Model can be expressed using XML. A pseudo-example is shown in Example 2 in which permissions for display, print (with a constraint), and annotate have been granted.

```
<permission>
    <display/>
    <print>
        <constraint> ... </constraint>
    </print>
    <annotate/>
</permission>
```

Example 2.Permission Model XML Syntax

Complete and formal syntactical examples are given in Section 4 "ODRL XML Syntax".

2.3 ODRL Constraint Model

ODRL supports the expression of Rights Constraints. This is the recognised set of restrictions on the Permissions over the Asset. The ***ODRL*** Constraint Model is shown in Figure 3.

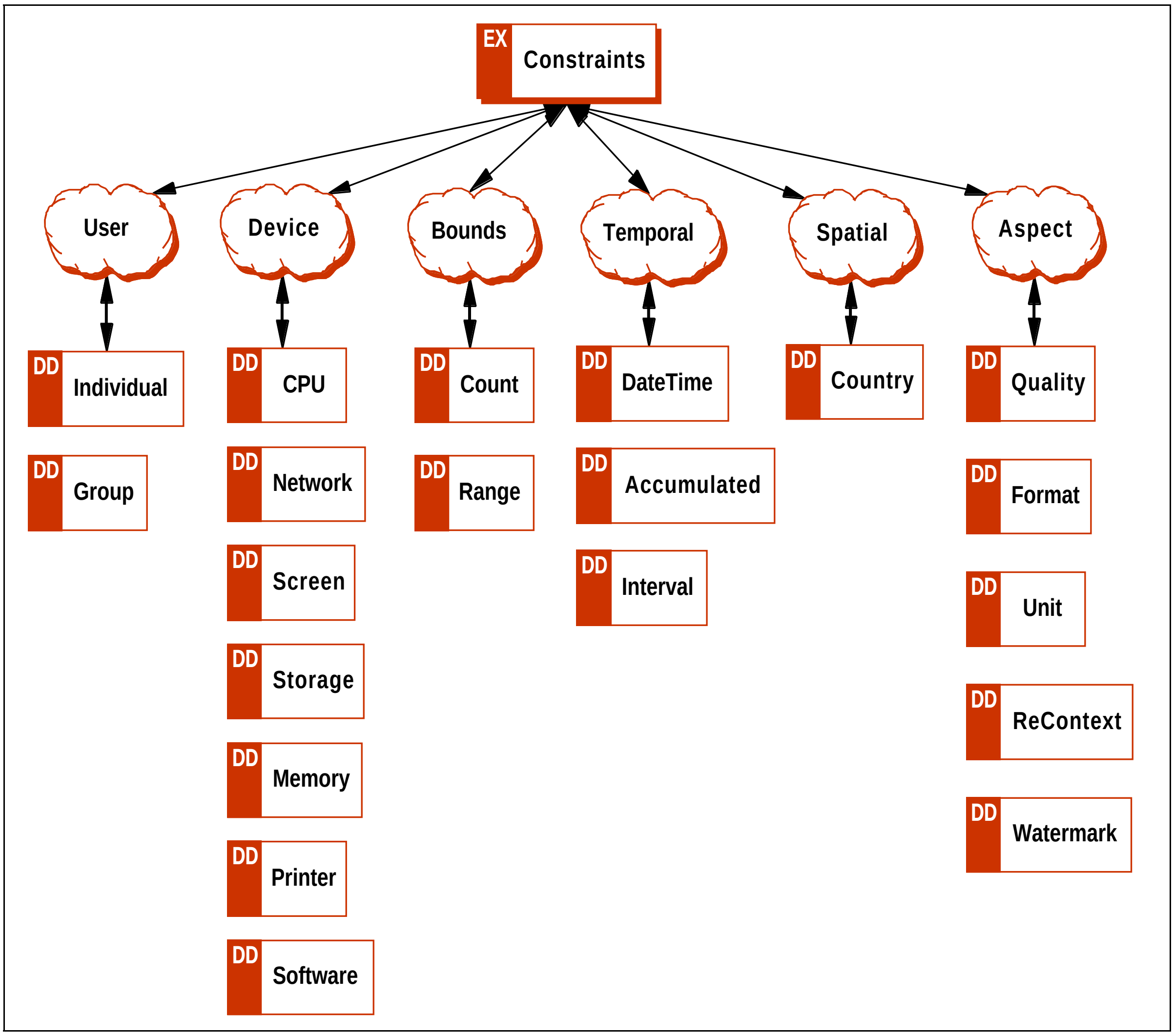

Figure 3. ODRL Constraint Model

The Constraints entity consists of an aggregation of six abstract entities. The abstract entities (depicted as clouds in Figure 3) are solely used to group similar constraints, and include:

- User - indicates a set of constraints which limits usage to identified user(s) (realised with: Individual, Group).
- Device - indicates a set of constraints which limits usage to physical devices or systems (realised with: CPU, Network, Screen, Storage, Memory, Printer, Software).
- Bounds - indicates a set of constraints which limits usage to a fixed number or extent (realised with: Count, Range).
- Temporal - indicates a set of constraints which limits usage to temporal boundaries (realised with: Date Time, Accumulated, Interval).
- Spatial - indicates a set of constraints which limits usage to spatial boundaries (realised with: Country).
- Aspect - indicates a set of constraints which limits usage to distinct features or expressions of the asset (realised with: Quality, Format, Unit, ReContext, Watermark).

Note: The listed elements above (eg Individual, Group etc) are defined in full in Section 3 "ODRL Data Dictionary Semantics" and form the basis of the ***ODRL*** Data Dictionary.

A Constraint is associated with one Permission. If a Constraint appears at the same level as a number of Permissions, then the Constraint applies to all of the Permissions. Most Constraints can also have zero or more other Constraints. However, the Bounds and Temporal constraints do not permit sub-constraints.

For Permissions with multiple constraints, all constraints must be honoured and no conflicts should arise. An error must be generated if the latter is true.

Important Note:

Any Constraint that is expressed but cannot be performed by the consuming system, must not be granted. That is, if a system does not understand how to guarantee that a specified constraint be honoured it must not grant the permission at all.

2.3.1 Constraint XML Example

The ***ODRL*** Constraint Model can be expressed using XML. A pseudo-example is shown in Example 3 in which the display permission is constrained to a particular CPU and may only be printed up to 5 times.

```
<display>
    <constraint>
        <cpu/>
    </constraint>
</display>
<print>
    <constraint>
        <count start=”1” end =”5”/>
    </constraint>
</print>
```

Example 3.Constraint Model XML Syntax

Complete and formal syntactical examples are given in Section 4 "ODRL XML Syntax".

2.4 ODRL Requirement Model

ODRL supports the expression of Rights Requirements. This is the recognised set of preconditions that must be met in order to obtain the related permissons. The ***ODRL*** Requirements Model is shown in Figure 4.

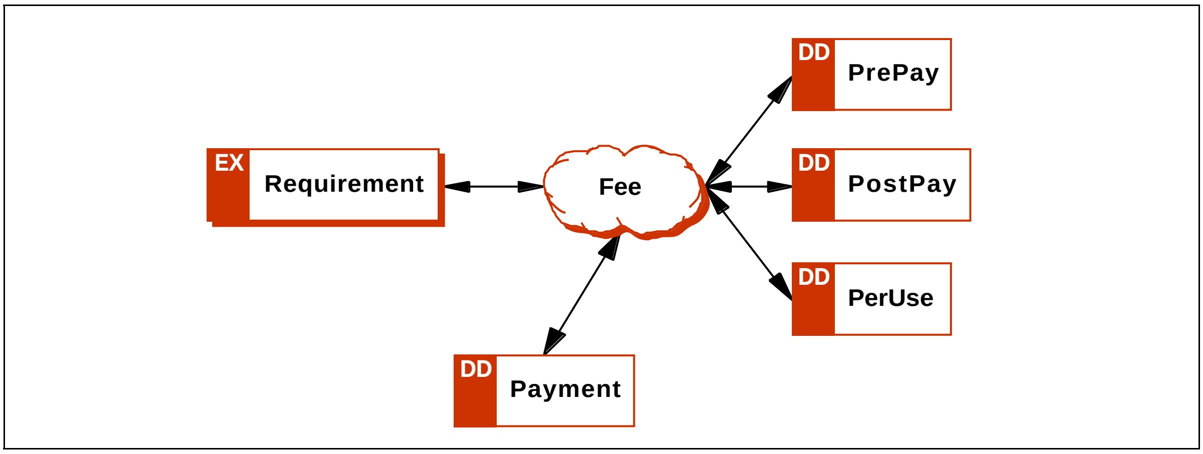

Figure 4. ODRL Requirement Model

The Requirement entity consists of one abstract entity. The abstract entity (depicted as a cloud in Figure 4) and is solely used to group similar Fee requirements, and includes:

- PrePay - indicates a payment amount that must be paid in advance of the use of the granted rights.
- PostPay - indicates a payment amount that can be paid after the use of the granted rights.
- PerUse - indicates a payment amount that must be paid each time the granted rights are consumed.

Note: The listed elements above (eg PrePay etc) are defined in full in Section 3 "ODRL Data Dictionary Semantics" and form the basis of the ***ODRL*** Data Dictionary.

A Requirement can be associated with one or many Permissions. If a Requirement appears at the same level as a number of Permissions, then the Requirement applies once to all of the Permissions. For Permissions with multiple Requirements, all Requirements must be honoured and no conflicts should arise. An error must be generated if the latter is true.

The ***ODRL*** Data Dictionary also defines the common reusable Payment entity, A payment consists of the amount to pay, the currency of the amount, the taxation due (as a percentage) and a taxation code.

Important Note:

Any Requirement that is expressed but cannot be performed by the consuming system, must not be granted. That is, if a system does not understand how to guarantee that a specified Requirement has been met, then it must not grant the permission at all.

2.4.1 Requirement XML Example

The ***ODRL*** Requirement Model can be expressed using XML. A pseudo-example is shown in Example 4 in which the play permission requires a $AUD20 fee (plus 10% tax) to be paid per use.

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

Example 4.Requirement Model XML Syntax

Complete and formal syntactical examples are given in Section 4 "ODRL XML Syntax".

2.5 ODRL Rights Holder Model

ODRL supports the identification of Rights Holders. The Rights Holder is a recognised Party and any set of entitlements that they are due for the use of their Asset. The ***ODRL*** Rights Holder Model is shown in Figure 5.

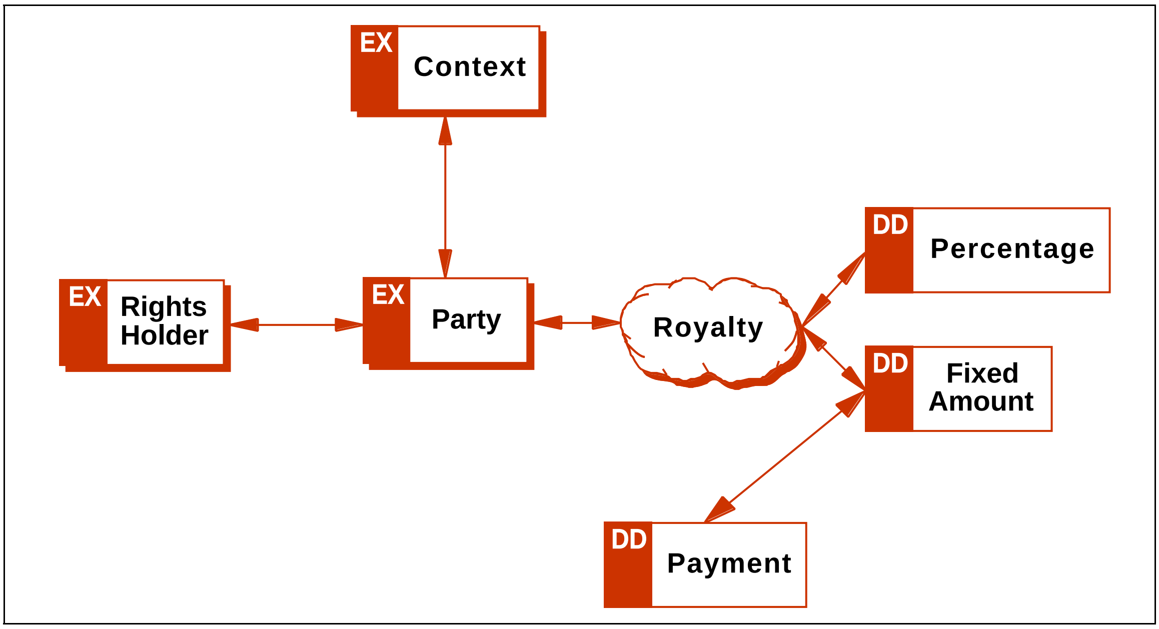

Figure 5. ODRL Rights Holder Model

The Rights Holder entity consists of one abstract entity: The abstract entity (depicted as a cloud in Figure 5) is solely used to group similar Rights Holder royalty entitlements, and include:

- Percentage - indicates a payment due to the indicated party for each transaction over the asset as a percentage of the value of the net transaction.
- Fixed Amount - indicates a payment due to the indicated party for each transaction over the asset as a fixed value of the net transaction.

Note: The listed elements above (eg Percentage etc) are defined in full in Section 3 "ODRL Data Dictionary Semantics" and form the basis of the ***ODRL*** Data Dictionary.

A Rights Holder can contain one or more Parties, and can be associated with one or many Assets. Parties can also be nested to indicate dependencies. For example; to support Rights Holder Sonya receiving 10% of Rights Holder Fiona’s royalty (who may receive a fixed amount).

2.5.1 Rights Holder XML Example

The ***ODRL*** RightsHolder Model can be expressed using XML. A pseudo-example is shown in Example 5 in which two identified parties share royalties with 90% to one and 10% to the other for each net transaction over their asset.

```
<rightsholder>
    <party>
        <context> ... </context>
        <percentage> 90 </percentage>
    </party>
    <party>
        <context> ... </context>
        <percentage> 10 </percentage>
    </party>
</rightsholder>
```

Example 5.RightsHolder Model XML Syntax

Complete and formal syntactical examples are given in Section 4 "ODRL XML Syntax".

2.6 ODRL Context Model

ODRL supports expressing additional information about entities and related entities. The ***ODRL*** Context Model is shown in Figure 6.

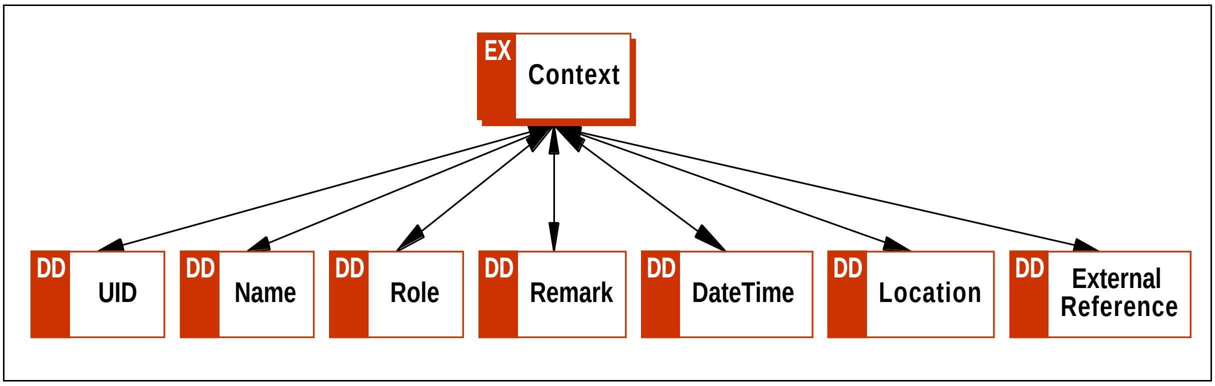

Figure 6. ODRL Context Model

The Context entity is an aggregation of seven other entities and includes:

- UID - a unique identification number for the entity.
- Name - a name used to describe the entity.
- Role - the role played by the entity.
- Remark - comments related to the entity.
- Date Time - indicates the date the event/entity occurred or is valid for.
- Location - indicates the physical location of the event/entity.
- External Reference - a link (URI) to an additional information about the entity.

Note: The listed elements above (eg UID etc) are defined in full in Section 3 "ODRL Data Dictionary Semantics" and form the basis of the ***ODRL*** Data Dictionary.

The Context is used for a number of different purposes and can be associated with any entity. When declaring Assets, it is used to indicate the unique identifier for the asset. When declaring Parties, it is used to indicate the unique identifier for the party, what role they may have played, and their name. The whole rights expression can also have a Context to indicate the unique identifier for the expression as well as when it was created. The Agreement entity can also have a Context to indicate the location of the transaction.

2.6.1 Context XML Example

The ***ODRL*** Context Model can be expressed using XML. A pseudo-example is shown in Example 6 in which the context for a party is described.

```
<party>
   <context>
      <uid idscheme=”X500”> c=EX;o=Federal Library;ou=Registry; cn=Maria K Brown </uid>
      <name> Maria Brown </name>
      <role idscheme=”ONIX”> AO1 </role>
      <reference> http://www.maria-k-brown.com/vcard.xml </reference>
   </context>
</party>
```

Example 6.Context Model XML Syntax

Complete and formal syntactical examples are given in Section 4 "ODRL XML Syntax".

2.7 ODRL Agreement Model

ODRL supports expressing agreements made between parties for specific rights over assets. The ***ODRL*** Agreement Model is shown in Figure 7.

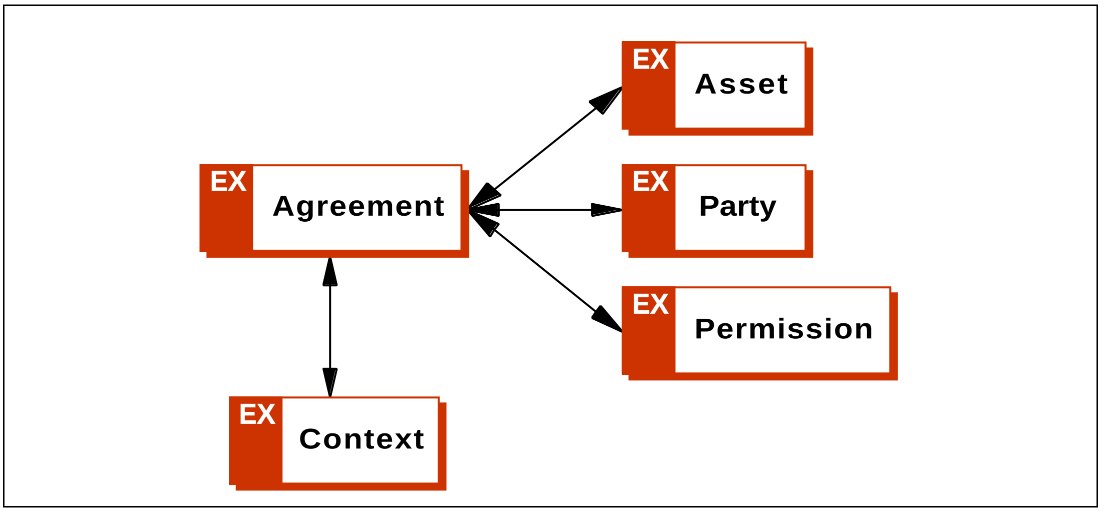

Figure 7. ODRL Agreement Model

The Agreement entity is an aggregation of four other entities and includes:

- Asset - information about the asset
- Party - information about the parties to the agreement.
- Permission - information (or a link) to the usage permissions being agreed to.
- Context - details about the agreement such as the date, time, location, identifier etc.

The Agreement entity allows for detailed expressions of particular parties who have negotiated and agreed to a set of particular permissions over some assets.

2.7.1 Agreement XML Example

The ***ODRL*** Agreement Model can be expressed using XML. A pseudo-example is shown in Example 7 in which an agreement, with context, is described between a party and a set of permissions over an asset.

```
<agreement>
    <context>
        <uid idscheme=”DOI”> 10.999/license/20010701/8736282828AAS </uid>
        <datetime> 2001-07-01T10:31:30 </datetime>
        <location> Sydney, Australia </location>
        <remark> Transacted by Example.Com</remark>
    </context>
    <party>
        <context> ... </context>
    </party>
    <asset> ... </asset>
    <permission>
         ...
    </permission>
</party>
```

Example 7.Agreement Model XML Syntax

Complete and formal syntactical examples are given in Section 4 "ODRL XML Syntax".

2.8 Expression Containers

The ***ODRL*** expression language also supports a number of grouping mechanisms for entities into containers. The groupings in the containers imply explicit semantics and must be supported.

There are three types of containers defined:

- And - the entities all must be supported / recognised
- Exclusive Or - only one of the entities must be supported / recognised
- Inclusive Or - one or more of the entities must be supported / recognised

A new structure, called “container” maybe utilised to aggregate other entities to enforce the above semantics. The container structure will have an attribute (“and”, “ex-or”, “in-or”) to indicate the container type. The default container type is “and”. Additionally, when no container structure is used, the default of “and” is implied.

The container semantics only applies to the immediate children of the container.

Typically the container structure is used to aggregate permissions, constraints, and requirements but can be used on other entities.

2.8.1 Container XML Example

The ***ODRL*** Container structure can be expressed using XML. A pseudo-example is shown in Example 8 in which the play permission is constrained to either (or both) the CPU of Storage device and the requirement is to either pay up front ($AUD200) or per use ($AUD1.50).

```
<permission>
    <play>
        <constraint>
            <container type="in-or">
                <cpu/>
                <storage/>
            </container>
        </constraint>
    </play>
    <requirement>
        <container type="ex-or">
            <prepay>
                <payment>
                    <amount currency=”AUD”> 200.00 </amount>
                </payment>
            </prepay>
            <peruse>
                <payment>
                    <amount currency=”AUD”> 1.50 </amount>
                </payment>
            </peruse>
        </container>
    </requirement>
</permission>
```

Example 8.Container XML Syntax

Complete and formal syntactical examples are given in Section 4 "ODRL XML Syntax".

2.9 Expression Linking

The ***ODRL*** expression language also supports the ability to explicitly link expression fragments. The links between the expression fragments imply explicit semantics and must be supported. The linking allows for ***ODRL*** expressions to be constructed by reusing existing expressions and allowing explicit links between fragments. Each ***ODRL*** fragment can be uniquely identified.

In the usual case, entities appearing in an ***ODRL*** rights expression is assumed to be related. That is, a Permission entity that appears with an Asset entity are assumed to be related (ie this expresses the Asset’s permissions). When explicit linking is used, then the link relationships take precedence. The examples in the next section show how these cases can be expressed.

2.9.1 Link XML Example

The ***ODRL*** Link structure can be expressed using XML. A pseudo-example is shown in Example 9 in which the sell permission only applies to ASSET-01 and the lend permission applies to both ASSET-01 and ASSET-02.

```
<rights>
    <asset id=”ASSET-01”>
        <context> ... <context>
    </asset>
    <asset id=”ASSET-02”>
        <context> ... <context>
    </asset>
    <permission>
        <asset idref=”ASSET-01”>
        <sell/>
    </permission>
    <permission>
        <asset idref=”ASSET-01”>
        <asset idref=”ASSET-02”>
        <lend/>
    </permission>
</rights>
```

Example 9.Link XML Syntax

Complete and formal syntactical examples are given in Section 4 "ODRL XML Syntax".

3 ODRL Data Dictionary Semantics

This section details the semantics of all the ***ODRL*** data dictionary elements that is used within the ***ODRL*** expression language. The ***ODRL*** data dictionary elements form the basis of the language and can be extended (see Section 5 "ODRL Extensibility") by additional elements. Each element in the data dictionary is defined using the following attributes defined by ISO-11179:

- Name - The label assigned to the data element
- Identifier - The unique identifier assigned to the data element (that is used for the syntactical encoding of the element)
- Definition - A statement that clearly represents the concept and essential nature of the data element
- Comment - A remark concerning the application of the data element

The data dictionary elements are defined below in the following five sections:

- Section 3.1 "Permissions Elements"
- Section 3.2 "Constraint Elements"
- Section 3.3 "Requirement Elements"
- Section 3.4 "Rights Holder Elements"
- Section 3.5 "Context Elements"

3.1 Permissions Elements

The Elements are defined in Table 1: Permission Elements.

Table 1. Permission Elements

Name	Identifier	Definition	Comment
Usage Permissons (pertaining to the end use of an asset)			
Display	display	The act of rendering the asset onto a visual device.	
Print	print	The act of rendering the asset onto paper or hard copy form.	
Play	play	The act of rendering the asset into audio/video form.	
Execute	execute	The act of rendering the asset into machine-readable form.	
Transfer Permissions (pertaining to the transfer rights of an asset)			
Sell	sell	The act of allowing the asset to be sold for exchange of value.	
Lend	lend	The act of allowing the asset to be available for temporary use then returned.	Temporal constraints are required.
Give	give	The act of allowing the asset to be given away (without exchange of value).	
Lease	lease	The act of allowing the asset to be available for periods of time for exchange of value.	Temporal constraints are required.

Table 1. Permission Elements

Name	Identifier	Definition	Comment
Reuse Permissions (pertaining to the re-utilisation of an asset)			
Modify	modify	The act of changing parts of the asset creating a new asset.	
Copy	copy	The act of extracting parts (or all) of the asset for reuse into another asset.	
Annotate	annotate	The act of adding notations/ commentaries to the asset creating a new asset.	

3.2 Constraint Elements

The Elements are defined in Table 2: Constraint Elements.

Table 2. Constraint Elements

Name	Identifier	Definition	Comment
User Constraints (Any human or organisation)			
Individual	individual	An identifiable party acting as an individual.	Use Context to identify the individual.
Group	group	A number of identifiable party acting as a collection of individuals.	Use Context to identify the group
Device Constraints (Any electronic or digital equipment or system)			
CPU	cpu	An identifiable system with a central processing unit (CPU).	Use Context to identify the device.
Network	network	An identifiable data network.	Use Context to identify the device. Use Range to indicate the IP Address restriction.
Screen	screen	An identifiable display output screen device.	For example, a screen reader or braille device. Use Context to identify the device.
Storage	storage	An identifiable storage media device.	For example, a hard disk or removable cartridge. Use Context to identify the device.
Memory	memory	An identifiable memory device.	For example, the clipboard. Use Context to identify the device.
Printer	printer	An identifiable hard copy printer.	Use Context to identify the device.
Software	software	An identifiable software application that must be present.	Use Context to identify the device.

Table 2. Constraint Elements

Name	Identifier	Definition	Comment
Bound Constraints (The numeric limits within which any entity can function)			
Count	count	A numeric count indicating the number of times the corresponding entity may be exercised. Contains the following: • start - the beginning of the count (inclusive) • end - the end of the count (inclusive)	Positive decimals must be supported. If there is no “start” or “end” value, then the count is open-ended. Note, “start” must always be less than or equal to “end” and one must always be present. For example, the Print usage may be constraint with a count of 1 to 10 meaning that the asset can be printed once or up to 10 times.
Range	range	A numeric range indicating the min / max values of the corresponding entity that the constraint applies to Contains the following: • min - the beginning of the range (inclusive) • max - the end of the range (inclusive)	Positive and negative decimals must be supported. If there is no “min” or “max” value, then the range is open-ended. Note: “min” must always be less than or equal to “max” and one must always be present. For example, this is used to specify that only pages numbered 1 to 10 may be printed (using the “pages” subunit entity).
Temporal Constraints (The time limits within which any entity can function)			
Date Time	datetime	A date / time-based range. Contains the following: • start - the beginning of the range (inclusive) • end - the end of the range (inclusive)	Date and Time value must conform to [ISO8601]. If there is no “start” and / or “end” value, then the range is open-ended. Note: “start” must always be less than or equal to “end” and one must always be present.
Accumulated	accumulated	The maximum period of metered usage time.	Period value must conform to [ISO8601]. For example “P30H” indicates a 30 hour period.
Interval	interval	Recurring period of time in which the rights can be exercised	Date and Time value must conform to [ISO8601]. For example “P7D” indicates a 7 day period.
Spatial Constraints (Any geographical range or extent)			
Country	country	Specification of a Country code	Use Context to identify the country with the codes specified by the [ISO3166] Scheme

Table 2. Constraint Elements

Name	Identifier	Definition	Comment
Aspect Constraints (Any distinct feature of the Asset)			
Quality	quality	Specification of constraints on the quality aspects of the asset. Contains the following: • type - the classification of the quality type	The values for the type attribute must be from a well known vocabulary and the source clearly identified. For example, the resolution of an image or number of colours.
Format	format	Specification of constraints on the format(s) of the asset. Contains the following: • type - the classification of the format type	The values for the type attribute must be from a well known vocabulary and the source clearly identified. For example, values can taken from the Internet Media Type [IMT] list.
Unit	unit	Specification of constraints on the whole asset or sub-parts of the asset. Contains the following: • type - the classification of the sub-unit part type	The values for the type attribute must be from a well known vocabulary and the source clearly identified.
ReContext	recontext	Specification if the asset may or may not be re-contextualised.	Boolean value.
Water Mark	watermark	Specification of watermarking requirements for the asset.	Use Context to identify the watermark information.

3.3 Requirement Elements

The Elements are defined in Table 3: Requirement Elements.

Table 3. Requirement Elements

Name	Identifier	Definition	Comment
Common Entities			
Payment	payment	The amount of the payment. Contains the following: • amount and currency • taxpercent and code	The amount must be a positive decimal to two decimal places. The mandatory currency must use [ISO4217] codes. The taxpercent must be a positive decimal between 0 and 100 (inclusive). The optional code can be any standard tax identifier.
Payment Requirements (Obligations in the form of financial payments)			
Pre Pay	prepay	The amount due prior to the granting / use of the rights.	Use Payment entity.
Post Pay	postpay	The amount due after the use of the rights.	Use Payment entity.
Per Use	peruse	The amount due for each use of the granted rights.	Use Payment entity.

3.4 Rights Holder Elements

The Elements are defined in Table 4: Rights Holder Elements.

Table 4. Rights Holder Elements

Name	Identifier	Definition	Comment
Royalty (Financial payments to Rights Holders)			
Percentage	percentage	The percentage of the net transaction amount due to the rights holder.	The value must be a positive decimal between 0 and 100 (inclusive). The total of all rights holders must not exceed 100.
Fixed Amount	fixedamount	The fixed amount due to the rights holder for each net transaction amount.	Use Payment entity.

3.5 Context Elements

The Elements are defined in Table 5: Context Elements.

Table 5. Context Elements

Name	Identifier	Definition	Comment
UID	uid	The unique identifier for the entity. Contains the following: • idscheme - indicates the scheme used for the identifier.	Valid idschemes include: • URI • DOI • X500 • ISBN • ISTC • ISO3166 • FPI • GUID
Name	name	The general name given to the entity.	
Role	role	The role played by the identified entity. Contains the following: • idscheme - indicates the scheme used for the identifier.	The role values may be selected from existing vocabulary schemes, including • MARC • ONIX • MPEG7
Remark	remark	General comments or description about the entity	
Date Event	dateevent	A date/time value for the entity. Contains the following: • date - the date/range of the event • event - the type of event/range	Date and Time value must conform to [ISO8601].
Location	location	The place where the entity occurred or is located.	
External Reference	reference	A link to additional information about the entity.	The link must be a URI and resolve to (at least) well formed XML.

4 ODRL XML Syntax

ODRL is expressed in schema-valid XML syntax. This syntax is formally specified by the XML Schemas defined in Appendix A and Appendix B. These are the normative references for the ***ODRL*** expression language and data dictionary (respectively).

Note: If the human language needs to be specified for any elements containing string values, then the use of the standard “xml:lang” attribute is recommended.

4.1 ODRL XML Schemas

ODRL utilises two XML Schemas. One schema defines the Expression Language elements and constructs (see Appendix A), the other defines the Data Dictionary elements (see Appendix B). Both must be used to support valid ***ODRL*** expressions. Further, the Data Dictionary schema is dependent on the Expression Language schema as the former defines elements that are constrained by the expression language model.

The Expression Language schema defines abstract elements that are utilised as the “place holder” for data dictionary elements. For example, the Expression Language schema defines:

```
<xsd:element name=”permissionElement” type=”o-ex:permissionType” abstract=”true”/>
```

which allows permission elements to be defined in the Data Dictionary schema, such as:

```
<xsd:element name=”display” substitutionGroup=”o-ex:permissionElement”/>
```

The Data Dictionary schema utilises the “substitutionGroup” mechanism from XML Schema to ensure that only the appropriate elements can be used in the correct position in the Expression Language.

4.2 ODRL XML Namespaces

ODRL supports XML Namespaces to indicate the scope and identity of its elements and other content description elements.

The version 0.9 XML Namespace URI for the ***ODRL*** Expression Language is:

http://odrl.net/0.9/ODRL-EX

The version 0.9 XML Namespace URI for the ***ODRL*** Data Dictionary is:

http://odrl.net/0.9/ODRL-DD

NOTE: These URIs should be considered *experimental* until the ***ODRL*** specification is formalised by an appropriate body and new XML Namespaces and URIs are assigned.

4.3 ODRL Linking

ODRL uses XML Schema ID and IDREF to refer from XML fragments to other fragments as described in Section 2.9 "Expression Linking". This is used to express the relationship between the core ***ODRL*** entities such as Asset, Permission, and Agreement. Such elements can be identified with the “id” attribute then referred to via “idref” attribute.

It is important to recognise that as the ***ODRL*** expressions become more complicated, the need to partition and express linkages becomes paramount in order to have manageable and reusable rights expressions. The linking mechanism allows for quite complex expressions to be generated whilst preserving the interpretability of the overall rights language.

4.4 ODRL XML Examples

The XML syntax will be explained via a serious of scenarios covering different content sectors (ebooks, video, education).

4.4.1 Ebook Scenario #1

Corky Rossi (an author) and Addison Rossi (an illustrator) publish their ebook via “EBooksRUS Publishers”. They wish to allow consumers to purchase (a requirement to pay $AUD20.00 plus 10% tax) the ebook which is restricted to a single CPU only and they are allowed to print a maximum of 2 copies. They will also allow the first 5 pages (Units) of the ebook to be viewed online for free (no requirement). Note the use of the NumberOfPages entity from the ONIX namespace to indicate the unit type.

The revenue split is 60% to the Author, 10% to the Illustrator and 30% to the Publisher. Their identities are shown from an X.500 repository and their roles indicated using the ONIX and MARC terms.

The XML encoding of this scenario in ***ODRL*** is shown below:

```
<?xml version=”1.0” encoding=”UTF-8”?>
<o-ex:rights xmlns:o-ex=”http://odrl.net/0.9/ODRL-EX”
        xmlns:o-dd=”http://odrl.net/0.9/ODRL-DD”
        xmlns:onix=”http://www.editeur.org/onix/ReferenceNames”
        xmlns:xsi=”http://www.w3.org/2000/10/XMLSchema-instance”
        xsi:schemaLocation=”http://odrl.net/0.9/ODRL-DD http://odrl.net/0.9/ODRL-DD-09.xsd
            http://odrl.net/0.9/ODRL-EX  http://odrl.net/0.9/ODRL-EX-09.xsd”>

    <o-ex:asset>
        <o-ex:context>
            <o-dd:uid o-dd:idscheme=”DOI”>10.999999/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:display>
            <o-ex:constraint>
                <o-dd:cpu/>
            </o-ex:constraint>
        </o-dd:display>
        <o-dd:print>
            <o-ex:constraint>
                <o-dd:count o-dd:end=”2”/>
            </o-ex:constraint>
        </o-dd:print>
        <o-ex:requirement>
            <o-dd:prepay>
                <o-dd:payment>
                    <o-dd:amount o-dd:currency=”AUD”>20.00</o-dd:amount>
                    <o-dd:taxpercent o-dd:code=”GST”>10.00</o-dd:taxpercent>
                </o-dd:payment>
            </o-dd:prepay>
        </o-ex:requirement>
    </o-ex:permission>

    <o-ex:permission>
        <o-dd:display>
            <o-ex:constraint>
                <o-dd:unit o-dd:type=”onix:NumberOfPages”>
                    <o-ex:constraint>
```

```
                            <o-dd:range o-dd:min=”1” o-dd:max=”5”/>
                        </o-ex:constraint>
                    </o-dd:unit>
                </o-ex:constraint>
            </o-dd:display>
        </o-ex:permission>

        <o-ex:rightsholder>
            <o-ex:party>
                <o-ex:context>
                    <o-dd:uid o-dd:idscheme=”X500”>c=AU;o=Rights Dir;cn=Corky Rossi</o-dd:uid>
                    <o-dd:role o-dd:idscheme=”ONIX”>A01</o-dd:role>
                </o-ex:context>
                <o-dd:percentage>60</o-dd:percentage>
            </o-ex:party>
            <o-ex:party>
                <o-ex:context>
                    <o-dd:uid o-dd:idscheme=”X500”>c=AU;o=Rights Dir;cn=Addison Rossi</o-dd:uid>
                    <o-dd:role o-dd:idscheme=”ONIX”>A12</o-dd:role>
                </o-ex:context>
                <o-dd:percentage>10</o-dd:percentage>
            </o-ex:party>
            <o-ex:party>
                <o-ex:context>
                    <o-dd:uid o-dd:idscheme=”X500”>c=AU;o=Rights Dir;cn=EBooksRUS</o-dd:uid>
                    <o-dd:role o-dd:idscheme=”MARC”>pbl</o-dd:role>
                </o-ex:context>
                <o-dd:percentage>30</o-dd:percentage>
            </o-ex:party>
        </o-ex:rightsholder>

    </o-ex:rights>
```

4.4.2 Ebook Scenario #2

Following from Ebook Scenario #1 (above), a consumer (Mary Smith) decides to purchase the ebook under the conditions outlined. The ***ODRL*** expression below shows the Agreement generated. The Agreement has a context (with identifier and date). The Permission now shows the details of the constraint that is specific to the consumer. (In this case, the CPU identifier is saved with the license.)

The XML encoding of this scenario in ***ODRL*** is shown below:

```
    <?xml version=”1.0” encoding=”UTF-8”?>
    <o-ex:rights xmlns:o-ex=”http://odrl.net/0.9/ODRL-EX”
            xmlns:o-dd=”http://odrl.net/0.9/ODRL-DD”
            xmlns:xsi=”http://www.w3.org/2000/10/XMLSchema-instance”
            xsi:schemaLocation=”http://odrl.net/0.9/ODRL-DD http://odrl.net/0.9/ODRL-DD-09.xsd
                http://odrl.net/0.9/ODRL-EX  http://odrl.net/0.9/ODRL-EX-09.xsd”>

        <o-ex:agreement>

            <o-ex:context>
                <o-dd:uid o-dd:idscheme=”DOI”>10.999999/license/1234567890-ABCDEF</o-dd:uid>
                <o-dd:dateEvent o-dd:date=”2001-06-01”/>
                <o-dd:location>Sydney, Australia</o-dd:location>
                <o-dd:remark>Transacted by Example.Com</o-dd:remark>
            </o-ex:context>

            <o-ex:asset>
                <o-ex:context>
                    <o-dd:uid o-dd:idscheme=”DOI”>10.999999/ebook/rossi-000001</o-dd:uid>
                </o-ex:context>
            </o-ex:asset>

            <o-ex:permission>
                <o-dd:display>
                    <o-ex:constraint>
                        <o-dd:cpu>
                            <o-ex:context>
                                <o-dd:uid o-dd:idscheme=”GUID”>
                                            WebBuy:CPD-ID:ER-393939-DSS-787878</o-dd:uid>
                            </o-ex:context>
                        </o-dd:cpu>
                    </o-ex:constraint>
                </o-dd:display>
                <o-dd:print>
                    <o-ex:constraint>
                        <o-dd:count o-dd:end=”2”/>
                    </o-ex:constraint>
                </o-dd:print>
                <o-ex:requirement>
                    <o-dd:prepay>
                        <o-dd:payment>
                            <o-dd:amount o-dd:currency=”AUD”>20.00</o-dd:amount>
                            <o-dd:taxpercent o-dd:code=”GST”>10.00</o-dd:taxpercent>
                        </o-dd:payment>
                    </o-dd:prepay>
                </o-ex:requirement>
            </o-ex:permission>

            <o-ex:party>
                <o-ex:context>
                    <o-dd:uid o-dd:idscheme=”DOI”>10.999999/users/msmth-0001111</o-dd:uid>
                    <o-dd:name>Mary Smith</o-dd:name>
                </o-ex:context>
            </o-ex:party>

        </o-ex:agreement>
    </o-ex:rights>
```

4.4.3 Ebook Scenario #3

The ebook for an “Electronic Book Exchange” voucher is entitled “XML: A Manager’s Guide”. The rights owner is Addison-Wesley.

There is an agreement with the Distributor of this book (a company called “XYZ”). They have rights to Sell up to 5000 copies of the book. The have “Narrow” rights for Sell.

Another licensed end user for this book is “John Doe”. All his permissions start from the beginning of 2001 and finish at the end of 2004. He has rights to view the book for a total of 30 days. He can print up to 5 copies on a “trusted printer”. He can print up to 5 pages between page 1 and 100 every week - up to a total of 100 pages - on an any printer. He can also extract 5000 bytes every week up to a total of 1,000,000 bytes onto the Clipboard (memory). He has a right to Give the book away after one year of the permissions starting.

Note the use of the NumberOfPages and NumbeOfBytes entities from the ONIX and EBX namespaces. Note the use of id and idref for indicating the relationships between the agreements / rightsholder and the asset.

The XML encoding of this scenario in ***ODRL*** is shown below:

```
<?xml version=”1.0” encoding=”UTF-8”?>
<o-ex:rights xmlns:o-ex=”http://odrl.net/0.9/ODRL-EX”
        xmlns:o-dd=”http://odrl.net/0.9/ODRL-DD”
        xmlns:onix=”http://www.editeur.org/onix/ReferenceNames”
        xmlns:ebx=”http://www.ebxwg.org/ebook/vocab/”
        xmlns:xsi=”http://www.w3.org/2000/10/XMLSchema-instance”
        xsi:schemaLocation=”http://odrl.net/0.9/ODRL-DD http://odrl.net/0.9/ODRL-DD-09.xsd
            http://odrl.net/0.9/ODRL-EX  http://odrl.net/0.9/ODRL-EX-09.xsd”>

    <o-ex:context>
        <o-dd:uid o-dd:idscheme=”DOI”>10.999999/voucher/2001/1234567890</o-dd:uid>
        <o-dd:dateEvent o-dd:date=”2001-05-01” o-dd:event=”issued”/>
    </o-ex:context>

    <o-ex:asset o-ex:id=”a001”>
        <o-ex:context>
            <o-dd:uid o-dd:idscheme=”ISBN”>872-2345-981</o-dd:uid>
            <o-dd:name>XML: A Manager’s Guide</o-dd:name>
        </o-ex:context>
    </o-ex:asset>

    <o-ex:rightsholder>
        <o-ex:asset o-ex:idref=”a001”/>
        <o-ex:party>
            <o-ex:context>
                <o-dd:uid o-dd:idscheme=”URI”>http://publishers.net/registry/AWL</o-dd:uid>
                <o-dd:name>Addison-Wesley</o-dd:name>
                <o-dd:reference>http://www.addison-wesley.com</o-dd:reference>
            </o-ex:context>
        </o-ex:party>
    </o-ex:rightsholder>

    <o-ex:agreement>
        <o-ex:asset o-ex:idref=”a001”/>
        <o-ex:party>
            <o-ex:context>
                <o-dd:uid o-dd:idscheme=”URI”>http://distributors.net/registry/xyz</o-dd:uid>
                <o-dd:name>XYZ Company</o-dd:name>
                <o-dd:role o-dd:idscheme=”MARC”>dst</o-dd:role>
            </o-ex:context>
        </o-ex:party>
        <o-ex:permission o-ex:narrow=”true”>
            <o-dd:sell>
                <o-ex:constraint>
                    <o-dd:count o-dd:end=”5000”/>
                </o-ex:constraint>
            </o-dd:sell>
        </o-ex:permission>
    </o-ex:agreement>
    <o-ex:agreement>
        <o-ex:asset o-ex:idref=”a001”/>
        <o-ex:party>
            <o-ex:context>
                <o-dd:uid o-dd:idscheme=”URI”>http://people.net/registry/john-doe-9999</o-dd:uid>
                <o-dd:name>John Doe</o-dd:name>
            </o-ex:context>
        </o-ex:party>
        <o-ex:permission>
            <o-ex:constraint>
                <o-dd:datetime o-dd:start=”2001-01-01” o-dd:end=”2004-12-31”/>
            </o-ex:constraint>
            <o-dd:display>
                <o-ex:constraint>
                    <o-dd:accumulated>P30D</o-dd:accumulated>
                </o-ex:constraint>
            </o-dd:display>
            <o-dd:print>
                <o-ex:container o-ex:type=”and”>
                    <o-ex:constraint>
                        <o-dd:count o-dd:end=”5”/>
                        <o-dd:printer>
                            <o-ex:context>
                                <o-dd:uid o-dd:idscheme=”GUID”>Trust:Print:4747474742222</o-dd:uid>
                            </o-ex:context>
                        </o-dd:printer>
                    </o-ex:constraint>
                    <o-ex:constraint>
                        <o-dd:unit o-dd:type=”onix:NumberOfPages”>
                            <o-ex:constraint>
                                <o-dd:count o-dd:end=”100”/>
                            </o-ex:constraint>
                        </o-dd:unit>
                        <o-dd:printer/>
                    </o-ex:constraint>
                    <o-ex:constraint>
                        <o-dd:unit o-dd:type=”onix:NumberOfPages”>
                            <o-ex:constraint>
                                <o-dd:range o-dd:min=”1” o-dd:max=”100”/>
                            </o-ex:constraint>
                        </o-dd:unit>
                        <o-dd:interval>P7D</o-dd:interval>
                        <o-dd:printer/>
                    </o-ex:constraint>
                </o-ex:container>
            </o-dd:print>
            <o-dd:copy>
                <o-ex:constraint>
                    <o-dd:memory>
                        <o-ex:constraint>
```

```
                            <o-dd:unit o-dd:type="ebx:NumberOfBytes">
                                <o-ex:constraint>
                                    <o-dd:range o-dd:min="1" o-dd:max="100"/>
                                </o-ex:constraint>
                            </o-dd:unit>
                            <o-dd:interval>P7D</o-dd:interval>
                            <o-dd:printer/>
                        </o-ex:constraint>
                    </o-dd:memory>
                </o-ex:constraint>
            </o-dd:copy>
            <o-dd:give>
                <o-ex:constraint>
                    <o-dd:datetime o-dd:start="2002-01-01"/>
                </o-ex:constraint>
            </o-dd:give>
        </o-ex:permission>

    </o-ex:agreement>
</o-ex:rights>
```

4.4.4 Video Scenario #1

A video has three rights holders. Massimo Canale receives 75% of the transactions and Simona Canale receives the other 25%. Additionally, Maria Canale receives 10% of Simona’s share. (Note the nesting of the last two parties.)

The video is offered at two different levels of quality (30 and 90dpi) and each has a different per use fee.

Note the use of the “resolution” entity from the MPEG7 namespace to indicate the Quality aspect of the asset.

The XML encoding of this scenario in ***ODRL*** is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<o-ex:rights xmlns:o-ex="http://odrl.net/0.9/ODRL-EX"
        xmlns:o-dd="http://odrl.net/0.9/ODRL-DD"
        xmlns:mpeg7="http://www.mpeg7.org/2001/MPEG-7_Schema"
        xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
        xsi:schemaLocation="http://odrl.net/0.9/ODRL-DD http://odrl.net/0.9/ODRL-DD-09.xsd
            http://odrl.net/0.9/ODRL-EX  http://odrl.net/0.9/ODRL-EX-09.xsd">

    <o-ex:asset>
        <o-ex:context>
            <o-dd:uid o-dd:idscheme="DOI">10.9999999/video/383838383</o-dd:uid>
            <o-dd:name>XML: The Movie</o-dd:name>
        </o-ex:context>
    </o-ex:asset>

    <o-ex:rightsholder>
        <o-ex:party>
            <o-ex:context>
                <o-dd:uid o-dd:idscheme="X500">c=IT;o=Registry;cn=Massimo Canale</o-dd:uid>
            </o-ex:context>
            <o-dd:percentage>75</o-dd:percentage>
        </o-ex:party>
        <o-ex:party>
            <o-ex:context>
                <o-dd:uid o-dd:idscheme=”X500”>c=IT;o=Registry;cn=Simona Canale</o-dd:uid>
            </o-ex:context>
            <o-dd:percentage>25</o-dd:percentage>
            <o-ex:party>
                <o-ex:context>
                    <o-dd:uid o-dd:idscheme=”X500”>c=IT;o=Registry;cn=Maria Canale</o-dd:uid>
                </o-ex:context>
                <o-dd:percentage>10</o-dd:percentage>
            </o-ex:party>
        </o-ex:party>
    </o-ex:rightsholder>

    <o-ex:permission>
        <o-dd:play>
            <o-ex:constraint>
                <o-dd:quality o-dd:type=”mpeg7:resolution”>
                    <o-ex:constraint>
                        <o-dd:range o-dd:max=”30”/>
                    </o-ex:constraint>
                </o-dd:quality>
            </o-ex:constraint>
            <o-ex:requirement>
                <o-dd:peruse>
                    <o-dd:payment>
                        <o-dd:amount o-dd:currency=”ITL”>1000</o-dd:amount>
                    </o-dd:payment>
                </o-dd:peruse>
            </o-ex:requirement>
        </o-dd:play>
        <o-dd:play>
            <o-ex:constraint>
                <o-dd:quality o-dd:type=”mpeg7:resolution”>
                    <o-ex:constraint>
                        <o-dd:range o-dd:max=”90”/>
                    </o-ex:constraint>
                </o-dd:quality>
            </o-ex:constraint>
            <o-ex:requirement>
                <o-dd:peruse>
                    <o-dd:payment>
                        <o-dd:amount o-dd:currency=”ITL”>5000</o-dd:amount>
                    </o-dd:payment>
                </o-dd:peruse>
            </o-ex:requirement>
        </o-dd:play>
    </o-ex:permission>
</o-ex:rights>
```

4.4.5 Education Scenario #1

An online lecture (software application) is offered (execute) for a per-use fee of $AUD2.00 which must be constrained to a certain network. Additionally, for a fee of $AUD1000.00 the lecture can be lent to (up to) 100 individuals.

The rights holder receives 50% of the execute fee and 25% of the lend fee. (Note the linking between the party entity and the permission entities.)

The XML encoding of this scenario in ***ODRL*** is shown below:

```
<?xml version=”1.0” encoding=”UTF-8”?>
<o-ex:rights xmlns:o-ex=”http://odrl.net/0.9/ODRL-EX”
        xmlns:o-dd=”http://odrl.net/0.9/ODRL-DD”
        xmlns:xsi=”http://www.w3.org/2000/10/XMLSchema-instance”
        xsi:schemaLocation=”http://odrl.net/0.9/ODRL-DD http://odrl.net/0.9/ODRL-DD-09.xsd
            http://odrl.net/0.9/ODRL-EX  http://odrl.net/0.9/ODRL-EX-09.xsd”>

    <o-ex:asset>
        <o-ex:context>
            <o-dd:uid o-dd:idscheme=”DOI”>10.9999999/learnobject/uni/aa/7373722</o-dd:uid>
            <o-dd:name>XML: The Lecture</o-dd:name>
        </o-ex:context>
    </o-ex:asset>

    <o-ex:rightsholder>
        <o-ex:party>
            <o-ex:context>
                <o-dd:uid o-dd:idscheme=”X500”>c=AU;o=UniA;cn=Prof Bean</o-dd:uid>
            </o-ex:context>
            <o-ex:container o-ex:type=”and”>
                <o-dd:percentage>50</o-dd:percentage>
                <o-ex:permission o-ex:idref=”p001”/>
            </o-ex:container>
            <o-ex:container o-ex:type=”and”>
                <o-dd:percentage>25</o-dd:percentage>
                <o-ex:permission o-ex:idref=”p002”/>
            </o-ex:container>
        </o-ex:party>
    </o-ex:rightsholder>

    <o-ex:permission o-ex:id=”p001”>
        <o-dd:execute>
            <o-ex:constraint>
                <o-dd:network/>
            </o-ex:constraint>
            <o-ex:requirement>
                <o-dd:peruse>
                    <o-dd:payment>
                        <o-dd:amount o-dd:currency=”AUD”>2.00</o-dd:amount>
                    </o-dd:payment>
                </o-dd:peruse>
            </o-ex:requirement>
        </o-dd:execute>
    </o-ex:permission>

    <o-ex:permission o-ex:id=”p002”>
        <o-dd:lend>
            <o-ex:constraint>
                <o-dd:individual>
                    <o-ex:constraint>
```

```
                        <o-dd:count o-dd:start=”1” o-dd:end=”100”/>
                    </o-ex:constraint>
                </o-dd:individual>
            </o-ex:constraint>
            <o-ex:requirement>
                <o-dd:prepay>
                    <o-dd:payment>
                        <o-dd:amount o-dd:currency=”AUD”>1000.00</o-dd:amount>
                    </o-dd:payment>
                </o-dd:prepay>
            </o-ex:requirement>
        </o-dd:lend>
    </o-ex:permission>

</o-ex:rights>
```

5 ODRL Extensibility

To be completed in the next revision (Version 0.95).

ODRL defines both the core structure of the expression language and a set of core elements as part of the ***ODRL*** data dictionary. Additional data dictionaries can be defined and used within the ***ODRL*** framework

Will include:

- Defining additional Data Dictionary elements
- Specifying equivalent rights
- Mapping between rights languages

6 Signing ODRL Expressions

To be completed in the next revision (Version 0.95).

Will include the use of XML Signatures.

7 Transporting ODRL Expressions

To be completed in the next revision (Version 0.95).

Will include the use of SOAP to transmit expressions.

8 References

Technical Standards:

[AAP] Digital Rights Management for Ebooks: Publisher Requirements, Version 1.0, Association of American Publishers, 2000.

<http://www.publishers.org/home/drm.pdf>

[DCMI] Dublin Core Metadata Initiative

<http://dublincore.org>

[DIG35] DIG35 Specification – Metadata for Digital Images – Version 1.0, August 30, 2000

<http://www.digitalimaging.org>

[DOI] Digital Object Identifier

<http://www.doi.org/>

[EBX] Electronic Book Exchange

<http://www.ebxwg.org/>

[FPI] Information Technology SGML Support Facilities Registration Procedures for Public Text Owner Identifiers, ISO/IEC 9070, Second Edition, International Organization for Standardization, April, 1991.

[GUID] Globally Unique Identifier

[IFLA] Functional Requirements for Bibliographic Records

<http://www.ifla.org/VII/s13/frbr/frbr.htm>

[IMS] IMS Global Learning Consortium

<http://www.imsproject.org/>

[IMT] Internet Media Types

<http://www.isi.edu/in-notes/iana/assignments/media-types/media-types>

[INDECS] Interoperability of Data in Ecommerce Systems

<http://www.indecs.org/>

[ISBN] International Standard Book Number

[ISTC] International Standard Textual Work Code

<http://www.nlc-bnc.ca/iso/tc46sc9/istc.htm>

[ISO3166] Country Names and Code Elements

<http://www.din.de/gremien/nas/nabd/iso3166ma/codlstp1/>

[ISO4217] Currency Names

<http://www.xe.net/gen/iso4217.htm>

[ISO8601] ISO (International Organization for Standardization). Representations of dates and times

<http://www.iso.ch/markete/8601.pdf>

[MARC] MARC Code List for Relators

<http://www.loc.gov/marc/relators/re0002r1.html>

[MPEG] Moving Picture Experts Group (WG 4,7,21)

<http://www.cselt.it/leonardo/mpeg/>

[MPEG7] MPEG7 Roles List

<http://www.mpeg7.org/2001/MPEG-7_Schema>

[OEBF] OpenEBook Forum

<http://www.openebook.org/>

[ONIX] ONIX International

<http://www.editeur.org/onix.html>

[PRISM] Publishing Requirements for Industry Standard Metadata, 2001

<http://prismstandard.org>

[PROPAGATE] Propagate Project

<http://www.propagate.net/>

[RFC2119] Key words for use in RFCs to Indicate Requirement Levels

<http://www.ietf.org/rfc/rfc2119.txt>

[URI] Uniform Resource Identifiers (URI): Generic Syntax

<http://www.ietf.org/rfc/rfc2396.txt>

[VCARD] vCard MIME Directory Profile

<http://www.ietf.org/rfc/rfc2426.txt>

[X500] Technical Overview of Directory Services Using the X.500 Protocol, 1992

<http://www.ietf.org/rfc/rfc1309.txt>

[XML] Extensible Markup Language 1.0

<http://www.w3.org/TR/REC-xml>

[XML NAMESPACE] Namespaces in XML

<http://www.w3.org/TR/REC-xml-names/>

[XML SCHEMA] XML Schema Part 1: Structures, Part 2: Datatypes

<http://www.w3.org/TR/xmlschema-1/>
<http://www.w3.org/TR/xmlschema-2/>

Papers:

[ERICKSON] A Digital Object Approach to Interoperable Rights Management, John S Erickson, D-Lib Magazine, June 2001, Volume 7 Number 6

<http://www.dlib.org/dlib/june01/erickson/06erickson.html>
<DOI:10.1045/june2001-erickson>

[HIGGS] Rights Management: Managing the Layers of Rights and Roles in the Knowledge Based Economy, Peter Higgs, 2000, IPR Systems Report.

<http://www.iprsystems.com/assets/0.2_Rights_Management.pdf>

[IANNELLA] Digital Rights Management (DRM) Architectures, Renato Iannella, D-Lib Magazine, June 2001, Volume 7 Number 6

<http://www.dlib.org/dlib/june01/iannella/06iannella.html>
<DOI:10.1045/june2001-iannella>

9 Acknowledgements

The editors wish to acknowledge contributions and feedback to this document from:

- Peter Higgs, IPR Systems
- David Parrott, Reuters
- Robert Bolick, McGraw-Hill
- Derek Renouf, Adaptive Arts

Appendix A: ODRL Expression Language XML Schema (Normative)

The version 0.9 XML Namespace URI for the ***ODRL*** Expression Language is:

http://odrl.net/0.9/ODRL-EX

The actual location of the XML Schema is:

http://odrl.net/0.9/ODRL-EX-09.xsd

The XML Schema is documented at:

http://odrl.net/0.9/ODRL-EX-09-DOC/index.html

```
<?xml version=”1.0” encoding=”UTF-8”?>
<xsd:schema targetNamespace=”http://odrl.net/0.9/ODRL-EX”
        xmlns:o-ex=”http://odrl.net/0.9/ODRL-EX”
        xmlns:xsd=”http://www.w3.org/2000/10/XMLSchema”
        elementFormDefault=”qualified” attributeFormDefault=”qualified” version=”0.9”>
    <xsd:element name=”rights” type=”o-ex:rightsType”/>
    <xsd:element name=”party”>
        <xsd:complexType>
            <xsd:choice minOccurs=”0” maxOccurs=”unbounded”>
                <xsd:element ref=”o-ex:context” minOccurs=”0”/>
                <xsd:element ref=”o-ex:rightsHolderElement” minOccurs=”0”/>
                <xsd:element ref=”o-ex:party” minOccurs=”0” maxOccurs=”unbounded”/>
                <xsd:element ref=”o-ex:container” minOccurs=”0” maxOccurs=”unbounded”/>
            </xsd:choice>
            <xsd:attributeGroup ref=”o-ex:IDGroup”/>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name=”permission”>
        <xsd:complexType>
            <xsd:complexContent>
                <xsd:extension base=”o-ex:permissionType”>
                    <xsd:attribute name=”exclusive” type=”xsd:boolean” use=”optional”/>
                    <xsd:attribute name=”narrow” type=”xsd:boolean” use=”optional”/>
                    <xsd:attributeGroup ref=”o-ex:IDGroup”/>
                </xsd:extension>
            </xsd:complexContent>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name=”constraintElement” type=”o-ex:constraintType” abstract=”true”/>
    <xsd:element name=”requiremetElement” type=”o-ex:requirementType” abstract=”true”/>
    <xsd:element name=”rightsHolderElement” type=”o-ex:rightsHolderType” abstract=”true”/>
    <xsd:element name=”permissionElement” type=”o-ex:permissionType” abstract=”true”/>
    <xsd:element name=”contextElement” type=”o-ex:contextType” abstract=”true”/>
    <xsd:element name=”context”>
        <xsd:complexType>
            <xsd:complexContent>
                <xsd:extension base=”o-ex:contextType”>
                    <xsd:attributeGroup ref=”o-ex:IDGroup”/>
                </xsd:extension>
            </xsd:complexContent>
        </xsd:complexType>
```

```
</xsd:element>
<xsd:complexTypename=”rightsType”>
    <xsd:choice maxOccurs=”unbounded”>
        <xsd:element ref=”o-ex:asset” maxOccurs=”unbounded”/>
        <xsd:element ref=”o-ex:permission” maxOccurs=”unbounded”/>
        <xsd:element name=”agreement” minOccurs=”0” maxOccurs=”unbounded”>
            <xsd:complexType>
                <xsd:complexContent>
                    <xsd:extension base=”o-ex:agreementType”>
                        <xsd:attributeGroup ref=”o-ex:IDGroup”/>
                    </xsd:extension>
                </xsd:complexContent>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name=”rightsholder” minOccurs=”0” maxOccurs=”unbounded”>
            <xsd:complexType>
                <xsd:complexContent>
                    <xsd:extension base=”o-ex:rightsHolderType”>
                        <xsd:attributeGroup ref=”o-ex:IDGroup”/>
                    </xsd:extension>
                </xsd:complexContent>
            </xsd:complexType>
        </xsd:element>
        <xsd:element ref=”o-ex:context” minOccurs=”0”/>
    </xsd:choice>
</xsd:complexType>
<xsd:element name=”container”>
    <xsd:complexType>
        <xsd:complexContent>
            <xsd:extension base=”o-ex:containerType”>
                <xsd:attribute name=”type” use=”required” value=”and”>
                    <xsd:simpleType>
                        <xsd:restriction base=”xsd:NMTOKEN”>
                            <xsd:enumeration value=”and”/>
                            <xsd:enumeration value=”in-or”/>
                            <xsd:enumeration value=”ex-or”/>
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:attribute>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
</xsd:element>
<xsd:complexTypename=”containerType”>
    <xsd:choice minOccurs=”0” maxOccurs=”unbounded”>
        <xsd:element ref=”o-ex:container” minOccurs=”0” maxOccurs=”unbounded”/>
        <xsd:element ref=”o-ex:permission” minOccurs=”0” maxOccurs=”unbounded”/>
        <xsd:element ref=”o-ex:permissionElement” minOccurs=”0” maxOccurs=”unbounded”/>
        <xsd:element ref=”o-ex:constraintElement” minOccurs=”0” maxOccurs=”unbounded”/>
        <xsd:element ref=”o-ex:requiremetElement” minOccurs=”0” maxOccurs=”unbounded”/>
        <xsd:element ref=”o-ex:rightsHolderElement” minOccurs=”0” maxOccurs=”unbounded”/>
        <xsd:element ref=”o-ex:constraint” minOccurs=”0” maxOccurs=”unbounded”/>
    </xsd:choice>
</xsd:complexType>
```

```
<xsd:complexTypename=”contextType”>
    <xsd:choice minOccurs=”0” maxOccurs=”unbounded”>
        <xsd:element ref=”o-ex:contextElement” minOccurs=”0” maxOccurs=”unbounded”/>
    </xsd:choice>
</xsd:complexType>
<xsd:complexTypename=”permissionType”>
    <xsd:choice minOccurs=”0” maxOccurs=”unbounded”>
        <xsd:element ref=”o-ex:permissionElement” minOccurs=”0” maxOccurs=”unbounded”/>
        <xsd:element ref=”o-ex:container” minOccurs=”0” maxOccurs=”unbounded”/>
        <xsd:element name=”requirement” minOccurs=”0” maxOccurs=”unbounded”>
            <xsd:complexType>
                <xsd:complexContent>
                    <xsd:extension base=”o-ex:requirementType”>
                        <xsd:attributeGroup ref=”o-ex:IDGroup”/>
                    </xsd:extension>
                </xsd:complexContent>
            </xsd:complexType>
        </xsd:element>
        <xsd:element ref=”o-ex:constraint” minOccurs=”0” maxOccurs=”unbounded”/>
    </xsd:choice>
</xsd:complexType>
<xsd:complexTypename=”requirementType”>
    <xsd:sequence minOccurs=”0” maxOccurs=”unbounded”>
        <xsd:element ref=”o-ex:requiremetElement” minOccurs=”0” maxOccurs=”unbounded”/>
        <xsd:element ref=”o-ex:container” minOccurs=”0” maxOccurs=”unbounded”/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexTypename=”constraintType”>
    <xsd:choice minOccurs=”0” maxOccurs=”unbounded”>
        <xsd:element ref=”o-ex:constraintElement” minOccurs=”0” maxOccurs=”unbounded”/>
        <xsd:element ref=”o-ex:container” minOccurs=”0” maxOccurs=”unbounded”/>
    </xsd:choice>
</xsd:complexType>
<xsd:complexTypename=”assetType”>
    <xsd:choice minOccurs=”0” maxOccurs=”unbounded”>
        <xsd:element ref=”o-ex:context”/>
    </xsd:choice>
</xsd:complexType>
<xsd:complexTypename=”rightsHolderType”>
    <xsd:choice minOccurs=”0” maxOccurs=”unbounded”>
        <xsd:element ref=”o-ex:context” minOccurs=”0”/>
        <xsd:element ref=”o-ex:party” maxOccurs=”unbounded”/>
        <xsd:element ref=”o-ex:asset” minOccurs=”0”/>
    </xsd:choice>
</xsd:complexType>
<xsd:complexTypename=”agreementType”>
    <xsd:choice minOccurs=”0” maxOccurs=”unbounded”>
        <xsd:element ref=”o-ex:party” maxOccurs=”unbounded”/>
        <xsd:element ref=”o-ex:asset” maxOccurs=”unbounded”/>
        <xsd:element ref=”o-ex:permission” maxOccurs=”unbounded”/>
        <xsd:element ref=”o-ex:context” minOccurs=”0”/>
    </xsd:choice>
</xsd:complexType>
<xsd:element name=”asset”>
```

```
        <xsd:complexType>
            <xsd:complexContent>
                <xsd:extension base=”o-ex:assetType”>
                    <xsd:attributeGroup ref=”o-ex:IDGroup”/>
                    <xsd:attribute name=”type”>
                        <xsd:simpleType>
                            <xsd:restriction base=”xsd:NMTOKEN”>
                                <xsd:enumeration value=”work”/>
                                <xsd:enumeration value=”expression”/>
                                <xsd:enumeration value=”manifestation”/>
                                <xsd:enumeration value=”item”/>
                            </xsd:restriction>
                        </xsd:simpleType>
                    </xsd:attribute>
                </xsd:extension>
            </xsd:complexContent>
        </xsd:complexType>
    </xsd:element>
    <xsd:attributeGroup name=”IDGroup”>
        <xsd:attribute name=”id” type=”xsd:ID”/>
        <xsd:attribute name=”idref” type=”xsd:IDREF”/>
    </xsd:attributeGroup>
    <xsd:element name=”constraint”>
        <xsd:complexType>
            <xsd:complexContent>
                <xsd:extension base=”o-ex:constraintType”>
                    <xsd:attributeGroup ref=”o-ex:IDGroup”/>
                </xsd:extension>
            </xsd:complexContent>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Appendix B: ODRL Data Dictionary XML Schema (Normative)

The version 0.9 XML Namespace URI for the ***ODRL*** Data Dictionary is:

http://odrl.net/0.9/ODRL-DD

The actual location of the XML Schema is:

http://odrl.net/0.9/ODRL-DD-09.xsd

The XML Schema is documented at:

http://odrl.net/0.9/ODRL-DD-09-DOC/index.html

```
<?xml version=”1.0” encoding=”UTF-8”?>
<xsd:schema targetNamespace=”http://odrl.net/0.9/ODRL-DD”
    xmlns:xsd=”http://www.w3.org/2000/10/XMLSchema”
    xmlns:o-dd=”http://odrl.net/0.9/ODRL-DD”
    xmlns:o-ex=”http://odrl.net/0.9/ODRL-EX”
    elementFormDefault=”qualified” attributeFormDefault=”qualified” version=”0.9”>
  <xsd:import namespace=”http://odrl.net/0.9/ODRL-EX”
      schemaLocation=”http://odrl.net/0.9/ODRL-EX-09.xsd”/>
  <xsd:element name=”payment”>
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name=”amount”>
          <xsd:complexType>
            <xsd:simpleContent>
              <xsd:extension base=”xsd:decimal”>
                <xsd:attribute name=”currency” type=”xsd:NMTOKEN” use=”required”/>
              </xsd:extension>
            </xsd:simpleContent>
          </xsd:complexType>
        </xsd:element>
        <xsd:element name=”taxpercent” minOccurs=”0”>
          <xsd:complexType>
            <xsd:simpleContent>
              <xsd:extension base=”xsd:decimal”>
                <xsd:attribute name=”code” type=”xsd:NMTOKEN” use=”required”/>
              </xsd:extension>
            </xsd:simpleContent>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name=”display” substitutionGroup=”o-ex:permissionElement”/>
  <xsd:element name=”print” substitutionGroup=”o-ex:permissionElement”/>
  <xsd:element name=”play” substitutionGroup=”o-ex:permissionElement”/>
  <xsd:element name=”execute” substitutionGroup=”o-ex:permissionElement”/>
  <xsd:element name=”sell” substitutionGroup=”o-ex:permissionElement”/>
  <xsd:element name=”lend” substitutionGroup=”o-ex:permissionElement”/>
  <xsd:element name=”give” substitutionGroup=”o-ex:permissionElement”/>
  <xsd:element name=”lease” substitutionGroup=”o-ex:permissionElement”/>
  <xsd:element name=”modify” substitutionGroup=”o-ex:permissionElement”/>
```

```
    <xsd:element name=”copy” substitutionGroup=”o-ex:permissionElement”/>
    <xsd:element name=”annotate” substitutionGroup=”o-ex:permissionElement”/>
    <xsd:element name=”prepay” type=”o-dd:feeType” substitutionGroup=”o-ex:requiremetElement”/
>
    <xsd:element name=”postpay” type=”o-dd:feeType” substitutionGroup=”o-
ex:requiremetElement”/>
    <xsd:element name=”peruse” type=”o-dd:feeType” substitutionGroup=”o-ex:requiremetElement”/
>
    <xsd:complexTypename=”feeType”>
        <xsd:sequence>
            <xsd:element ref=”o-dd:payment”/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name=”fixedamount” type=”o-dd:feeType” substitutionGroup=”o-
ex:rightsHolderElement”/>
    <xsd:element name=”percentage” substitutionGroup=”o-ex:rightsHolderElement”>
        <xsd:simpleType>
            <xsd:restriction base=”xsd:decimal”>
                <xsd:minInclusive value=”0”/>
                <xsd:maxInclusive value=”100”/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:element>
    <xsd:element name=”uid” substitutionGroup=”o-ex:contextElement”>
        <xsd:complexType>
            <xsd:simpleContent>
                <xsd:extension base=”xsd:string”>
                    <xsd:attribute name=”idscheme” use=”required”>
                        <xsd:simpleType>
                            <xsd:restriction base=”xsd:NMTOKEN”>
                                <xsd:enumeration value=”URI”/>
                                <xsd:enumeration value=”DOI”/>
                                <xsd:enumeration value=”X500”/>
                                <xsd:enumeration value=”ISBN”/>
                                <xsd:enumeration value=”ISTC”/>
                                <xsd:enumeration value=”ISO3166”/>
                                <xsd:enumeration value=”FPI”/>
                                <xsd:enumeration value=”GUID”/>
                            </xsd:restriction>
                        </xsd:simpleType>
                    </xsd:attribute>
                </xsd:extension>
            </xsd:simpleContent>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name=”name” type=”xsd:string” substitutionGroup=”o-ex:contextElement”/>
    <xsd:element name=”role” substitutionGroup=”o-ex:contextElement”>
        <xsd:complexType>
            <xsd:simpleContent>
                <xsd:extension base=”xsd:string”>
                    <xsd:attribute name=”idscheme” use=”optional”>
                        <xsd:simpleType>
                            <xsd:restriction base=”xsd:NMTOKEN”>
                                <xsd:enumeration value=”MARC”/>
                                <xsd:enumeration value=”ONIX”/>
                                <xsd:enumeration value=”MPEG7”/>
                            </xsd:restriction>
                        </xsd:simpleType>
                    </xsd:attribute>
                </xsd:extension>
            </xsd:simpleContent>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name=”remark” type=”xsd:string” substitutionGroup=”o-ex:contextElement”/>
    <xsd:element name=”dateEvent” substitutionGroup=”o-ex:contextElement”>
        <xsd:complexType>
            <xsd:attribute name=”date” type=”xsd:date”/>
            <xsd:attribute name=”event” type=”xsd:string”/>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name=”location” type=”xsd:string” substitutionGroup=”o-ex:contextElement”/>
    <xsd:element name=”reference” type=”xsd:uriReference” substitutionGroup=”o-
ex:contextElement”/>
    <xsd:element name=”individual” substitutionGroup=”o-ex:constraintElement”>
        <xsd:complexType>
            <xsd:choice minOccurs=”0” maxOccurs=”unbounded”>
                <xsd:element ref=”o-ex:constraint”/>
                <xsd:element ref=”o-ex:context” minOccurs=”0”/>
            </xsd:choice>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name=”group” substitutionGroup=”o-ex:constraintElement”>
        <xsd:complexType>
            <xsd:choice minOccurs=”0” maxOccurs=”unbounded”>
                <xsd:element ref=”o-ex:constraint”/>
                <xsd:element ref=”o-ex:context”/>
            </xsd:choice>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name=”cpu” substitutionGroup=”o-ex:constraintElement”>
        <xsd:complexType>
            <xsd:choice minOccurs=”0” maxOccurs=”unbounded”>
                <xsd:element ref=”o-ex:constraint”/>
                <xsd:element ref=”o-ex:context”/>
            </xsd:choice>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name=”network” substitutionGroup=”o-ex:constraintElement”>
        <xsd:complexType>
            <xsd:choice minOccurs=”0” maxOccurs=”unbounded”>
                <xsd:element ref=”o-ex:constraint”/>
                <xsd:element ref=”o-ex:context”/>
            </xsd:choice>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name=”screen” substitutionGroup=”o-ex:constraintElement”>
        <xsd:complexType>
            <xsd:choice minOccurs=”0” maxOccurs=”unbounded”>
                <xsd:element ref=”o-ex:constraint”/>
                <xsd:element ref=”o-ex:context”/>
            </xsd:choice>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name=”storage” substitutionGroup=”o-ex:constraintElement”>
        <xsd:complexType>
            <xsd:choice minOccurs=”0” maxOccurs=”unbounded”>
                <xsd:element ref=”o-ex:constraint”/>
                <xsd:element ref=”o-ex:context”/>
            </xsd:choice>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name=”memory” substitutionGroup=”o-ex:constraintElement”>
        <xsd:complexType>
            <xsd:choice minOccurs=”0” maxOccurs=”unbounded”>
                <xsd:element ref=”o-ex:constraint”/>
                <xsd:element ref=”o-ex:context”/>
            </xsd:choice>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name=”printer” substitutionGroup=”o-ex:constraintElement”>
        <xsd:complexType>
            <xsd:choice minOccurs=”0” maxOccurs=”unbounded”>
                <xsd:element ref=”o-ex:constraint”/>
                <xsd:element ref=”o-ex:context”/>
            </xsd:choice>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name=”software” substitutionGroup=”o-ex:constraintElement”>
        <xsd:complexType>
            <xsd:choice minOccurs=”0” maxOccurs=”unbounded”>
                <xsd:element ref=”o-ex:constraint”/>
                <xsd:element ref=”o-ex:context”/>
            </xsd:choice>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name=”count” substitutionGroup=”o-ex:constraintElement”>
        <xsd:complexType>
            <xsd:attribute name=”start” use=”optional”>
                <xsd:simpleType>
                    <xsd:restriction base=”xsd:decimal”>
                        <xsd:minInclusive value=”0”/>
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:attribute>
            <xsd:attribute name=”end” use=”optional”>
                <xsd:simpleType>
                    <xsd:restriction base=”xsd:decimal”>
                        <xsd:minInclusive value=”0”/>
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:attribute>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name=”range” substitutionGroup=”o-ex:constraintElement”>
        <xsd:complexType>
            <xsd:attribute name=”min” type=”xsd:decimal” use=”optional”/>
            <xsd:attribute name=”max” type=”xsd:decimal” use=”optional”/>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name=”datetime” substitutionGroup=”o-ex:constraintElement”>
        <xsd:complexType>
            <xsd:attribute name=”start” type=”xsd:date” use=”optional”/>
            <xsd:attribute name=”end” type=”xsd:date” use=”optional”/>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name=”accumulated” type=”xsd:duration” substitutionGroup=”o-
ex:constraintElement”/>
    <xsd:element name=”interval” type=”xsd:duration” substitutionGroup=”o-ex:constraintElement”/>
    <xsd:element name=”country” substitutionGroup=”o-ex:constraintElement”>
        <xsd:complexType>
            <xsd:choice minOccurs=”0” maxOccurs=”unbounded”>
                <xsd:element ref=”o-ex:constraint”/>
                <xsd:element ref=”o-ex:context”/>
            </xsd:choice>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name=”quality” substitutionGroup=”o-ex:constraintElement”>
        <xsd:complexType>
            <xsd:choice minOccurs=”0” maxOccurs=”unbounded”>
                <xsd:element ref=”o-ex:constraint”/>
            </xsd:choice>
            <xsd:attribute name=”type” type=”xsd:QName”/>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name=”format” substitutionGroup=”o-ex:constraintElement”>
        <xsd:complexType>
            <xsd:choice minOccurs=”0” maxOccurs=”unbounded”>
                <xsd:element ref=”o-ex:constraint”/>
            </xsd:choice>
            <xsd:attribute name=”type” type=”xsd:QName”/>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name=”unit” substitutionGroup=”o-ex:constraintElement”>
        <xsd:complexType>
            <xsd:choice minOccurs=”0” maxOccurs=”unbounded”>
                <xsd:element ref=”o-ex:constraint”/>
            </xsd:choice>
            <xsd:attribute name=”type” type=”xsd:QName”/>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name=”recontext” substitutionGroup=”o-ex:constraintElement”>
        <xsd:complexType>
            <xsd:choice minOccurs=”0” maxOccurs=”unbounded”>
                <xsd:element ref=”o-ex:constraint”/>
            </xsd:choice>
            <xsd:attribute name=”value” type=”xsd:boolean” use=”required”/>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name=”watermark” substitutionGroup=”o-ex:constraintElement”>
        <xsd:complexType>
            <xsd:choice minOccurs=”0” maxOccurs=”unbounded”>
                <xsd:element ref=”o-ex:context”/>
                <xsd:element ref=”o-ex:constraint”/>
            </xsd:choice>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```