Chapter 3

Rights Expression Languages (RELs)

This chapter provides an insight to the field of rights expression languages (RELs). It justifies the application of rights expression languages in rights expressions respectively electronic contracts in today’s DRM systems and addresses the requirements which have to be met by these languages (see Section 3.2). Section 3.3 addresses the language syntax and vocabulary of RELs. Standardisation is a critical success factor for RELs. If a REL has been accepted from or is supported by a standardising body, such as the W3C\(^1\) or the Open Mobile Alliance\(^2\), it is likely to become globally accepted respectively applied (as is exemplified with XML or RDF). Therefore, Section 3.4 introduces REL initiatives and their background, as well as languages that have already been accepted of standard-setting bodies. Practical examples (XML instances) of rights languages are given for the RELs ODRL and XrML. Finally, the chapter provides a short survey of the current market situation and trends in the field of DRM middleware and implementations using RELs (see Section 3.5).

3.1 Definition of Terms

**Rights Expression Language**: A rights expression language is a means of expressing usage and access rights of parties to assets. Rights

---

\(^1\)See: http://www.w3.org

\(^2\)See: http://www.openmobilealliance.org/
expression languages provide a syntax and semantics that are sufficiently rich to formulate rights expressions for digital publications, audio and video files, images, games, software, and other digital or physical goods, including pricing models as well as terms and conditions, regardless of whether a monetary consideration is part of the transaction. Consequently, rights expression languages provide metadata framework for the expression of rights.

**REL instance or rights expression:** Every document that is formulated in a rights expression language shall be defined as 'REL instance'. REL instances are exchanged between DRM systems respectively DRM system components and serve as interface between them. After being exchanged REL instances are further processed in applications, such as access control, accounting, etc. Sometimes the term rights expression is used as a synonym for REL instance. Depending on their content, rights expressions can represent different semantic constructs, e.g. licenses, digital tickets, or contracts (see Chapter 4, and in particular Section 4.3.1). A REL can be used to formulate "simple" rights expressions, such as "party X has the permission to play the resource Y", as well as complex electronic contracts where all contracting parties, the traded resources or rights, and the terms and conditions are specified in great detail. The mandatory elements in REL instances are defined in the respective REL specification. The difference between a "simple" rights expression and an electronic contract lies within content and its semantics. A contract requires at least the following elements: parties (consumer and rightsholder), resource, and permission.

### 3.2 Requirements of RELs

In order to provide a means of expression rights of parties to assets, a REL has to fulfill several technical and conceptual requirements. One substantial technical requirement of RELs is machine readability. Documents are machine readable, if a computer is able to digitally record the document information. Various techniques meet this requirement, e.g. even a newspaper article is machine readable after scanning the article and processing the image with an OCR (optical character recognition) software. OCR software, however, is not 100 percent reliable in terms of correct symbol recognition. Most of today's RELs are developed for the serialisation in XML, allowing for a formal representation of electronic contracts. Reading XML docu-
ments by a machine is more reliable than reading scanned documents and thus XML qualifies as an exchange format for rights expressions. XML is described in more detail in Section 3.4.1.

In Section 2.3.1 the function Content Consumption is described with the following sample DRM activities: handling the access request of the consumer, authenticating and identifying the consumer, identifying the resource that is subject to the access request, decrypting and decompressing the digital good(s), granting or denying access to the digital goods according to the license, and rendering the digital goods according to the granted permissions in the license. A number of REL requirements can be derived from these sample activities. In order to provide the relevant metadata, the REL should support

- **identification mechanisms**: unique ids are needed for the identification of parties (e.g. x500 [IT93a]) and resources (e.g. DOI [Nat00], ISBN [ISO92], ISSN [ISO98], etc.).

- **the definition of usage and access rights**: usage and access rights are e.g. play, print, copy, etc.

- **the definition of permission and duties**: permissions and duties, such as 'play Ebook no. 12356' or 'pay €100.00' are (operation, object) pairs where operation is an action (play) that may be or has to be performed on a certain resource.

- **the definition of constraints**: constraints are needed to narrow duties or usage and access rights in time, location, device, etc.

- **the articulation of roles**: some security mechanisms grant or deny access to resources depending the user's role rather than his/her identification number. Therefore, RELs also have to provide the possibility to express user roles, respectively express usage and access rights for roles.

- **the definition of technical details**: this supports the handling of decryption algorithms, viewers, etc.

- **workflow data**: this supports the course of the DRM process.

This informal enumeration does not represent a complete list of requirements for a REL. The Moving Picture Experts Group (MPEG) has specified...
the requirements for a rights expression language and its rights data dictionary in detail for the multimedia domain [Bor02]. The MPEG requirements comprise those listed above and a large number of additional ones, such as concepts for content aggregation, the sequencing of elements, etc. The work of Neal et al. [NCL+03] deals with the requirements for a special Business Contract Language. The definition of time constraints is addressed in detail, but the sequencing of operations (e.g. order, deliver, pay), and general constraints are only shortly sketched. The MPEG documents provide a comprehensive list of requirements and will be used as reference list for REL requirement in this thesis. The requirements of a rights language vary depending on their application field and scope, consequently RELs should be open and extensible.

3.3 Characteristics of RELs

Two basic factors in a language are its syntax and lexis (or vocabulary). The vocabulary of a language includes words that are created from permitted symbols (e.g. letters, numbers, and symbols). The syntax applies to the language vocabulary with which syntactically valid sentences can be formulated. Another crucial issue in language analysis is the field of semantics. The term semantics refers to the study of meaning as encoded in language [Wid96]. Syntax, lexis and the semantics of RELs are usually defined in a document called language specification [Ian02b, Con00, DWW03, Oct02]. The vocabulary of rights expression languages is sometimes referred to as rights data dictionary (RDD)) [Rig02, BR02]. With this specification an offer, contract, or other rights expression construct can be formulated.

Rights expression languages facilitate the interoperation of DRM systems and their components. They allow expressing rights information in a static format, i.e. putting down certain rights and conditions at a certain time/state. This static format provides

- The separation of space: Geographically distributed systems deploying different DRM technologies obtain means to communicate and interoperate.

- The separation of time: A contract is expressed at a certain state. At any time, business partners can re-read the contract and its conditions and verify its validity.
Rights expression languages have the potential to express aggregated rights information, i.e. RELs can phrase a good deal of rights information in a reduced representation. For example, permissions can be granted to groups of people (e.g. all students may access the script of informatics) or permissions can be granted to a type of resource that includes a large number of actual resources (e.g. user sguth may access all learning resources of the Department of Information Systems). Of course the software service that is processing such rights expression must be able to map the aggregated information to real objects.

3.3.1 REL Syntax

The basic elements in every REL syntax are permissions, resources and parties; the terminology for these three basic elements vary in each REL.

- **Permissions** are certain use or access rights to digital or physical goods or services. For the purpose of this thesis, *permission* is defined as operation–object pair. An operation is a certain action that can be performed on goods or services (objects), such as print, play, use, etc. Example for a permission is \((\text{print, test.pdf})\). Permissions can be specified in more detail by constraints. Constraints describe terms and conditions that have to be fulfilled before an operation is granted respectively serve to narrow the granted operation by time, location, individual, etc.

- The **resources** (or objects) represent the digital goods or services which the operations refer to. Resources have to be described by a non–ambiguous identifier such as DOI [Nat00].

- The **party** element represents any kind of party, i.e. a legal entity or a physical person which has a relationship (e.g. owns, controls, has permission to) to a digital product or service. In contracts, the party elements predominantly represent the people who enter into the contract. Examples of parties are the rights holder, the creator, the content provider, the consumer, the administrator, the beneficiary and the like.

Starting from these basic elements, each REL contains additional concepts for expressing containers, sequences, royalties, constraints, etc., and their relationships in more detail. Unfortunately, the REL community has not yet agreed on a general terminology for the basic REL elements. In one language the operation *play* is called a *right*, whereas in the other language
**play** is indicated as permission. Following the terminology of this thesis a permission is an operation–object pair. Due to the longer history and available research, this definition has been adopted from the access control community.

### 3.3.2 Rights Data Dictionary (RDD)

The rights data dictionary (vocabulary) of a REL defines the words that are permitted in REL instances and their semantics. For example, in a REL instance the terms print, play, or view may be used as operations and the terms time, location, and individual may be used to constraint permissions. The table below shows an extract from the ODRL Data Dictionary in which several operation elements are defined. Similar vocabulary definitions exist for other ODRL syntax elements, such as ODRL constraints, and the ODRL context element. Each term is usually defined by a name, an identifier, and a description. The description denotes the informal semantics to a certain term. The ODRL Data Dictionary is compliant to the ISO–11179 standard which provides naming and identification principles for data elements [ISO95].

<table>
<thead>
<tr>
<th>Name</th>
<th>Identifier</th>
<th>Description</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Play</td>
<td>play</td>
<td>The act of rendering the asset in audio/video form.</td>
<td>...</td>
</tr>
<tr>
<td>Print</td>
<td>print</td>
<td>The act of rendering the asset on paper or hard copy form.</td>
<td>...</td>
</tr>
<tr>
<td>Execute</td>
<td>execute</td>
<td>The act of executing the asset.</td>
<td>...</td>
</tr>
</tbody>
</table>

Other rights expression languages define their vocabulary in the same way (e.g. `<indecs>rdd [RB99]`) or similarly. XrML [Con00], for example, defines the informal, textual semantics of each lexical item in a small paragraph that also includes the respective extract of the XML schema defining the term. Additionally, the paragraph exemplifies the term’s usage, its relations to other terms, and exceptions. Therefore, the XrML specification version 2.0 is very complex. In contrast XrML, the clear RDD definition (due to the application of the ISO–11179 standard) in ODRL is more comprehensive. All languages introduced in this chapter allow for an extension of the RDD via XML subschemata.
RELs are often more powerful than the DRM system requires. Therefore, the rights expression language is usually adapted to the specific implementation and domain, i.e. a subset of the vocabulary or only a restricted syntax is used. For example, in the Colis\(^3\) project only a subset of access rights occurs in rights expressions [Ian03c]. For the purpose of this thesis, such adaptations are called *REL application policies*. Apart from defining the vocabulary subset such policies can also state the permitted identification schemes in instances (e.g. DOI, ISSN) or the depth of nested rights expressions. Application-specific rights expression generators and interpreters have to implement these policies.

### 3.4 Existing Rights Expression Languages and Initiatives

In this chapter, the most commonly used specifications in the field of rights expression languages are introduced. The field is still evolving, but the standards mentioned below have managed to prevail.

#### 3.4.1 Open Digital Rights Language (ODRL)

The Open Digital Rights Language (ODRL) [Ian02b] is being developed by the ODRL initiative\(^4\). The ODRL initiative is an international effort which aims at developing an open REL standard. In the spirit of the open source community, ODRL is freely available. It was recently accepted by the Open Mobile Alliance (OMA)\(^5\) as the standard REL for mobile content. The OMA aims at facilitating global user adoption of mobile data services. Therefore, OMA is developing specifications that ensure service interoperability across devices, geographies, service providers, operators, and networks, while allowing businesses to compete through innovation and differentiation. The latest version of the ODRL specification (version 1.1) has been co-published by W3C (as a W3C Note). The OpenIPMP Open Source Rights Management Project\(^6\) has just released the first version of their DRM software that utilises ODRL for formulating rights expressions. The ODRL initiative does not have a focus on some particular application domain. However, ODRL is well accepted in the telecommunication domain.

---

\(^3\)See: [http://www.colis.mq.edu.au/](http://www.colis.mq.edu.au/)

\(^4\)See: [http://www.odrl.net/](http://www.odrl.net/)


\(^6\)See: [http://www.openipmp.com/](http://www.openipmp.com/)
(through adoption of ODRL by OMA), in the educational domain, e.g. in the COLIS\(^7\) project, the learning federation\(^8\), the Open Archives Initiative [Bir01], and of the Dublin Core initiative [PJ02]. As ODRL is an open-source project, it is likely that the further development of ODRL will be research driven, i.e. researchers from all over the world will participate in future versions of ODRL. The release of ODRL version 2.0 is scheduled for the end of 2004. As a part of this thesis an interpreter for ODRL has been implemented. Therefore, ODRL will be discussed in more detail.

The following paragraphs present a REL syntax example of the straightforward concept of ODRL. Referring to the earlier definitions, an operation in ODRL, e.g. play or print, is called permission (!), resources are indicated in ODRL as assets, and parties are also called parties in ODRL. The root element in ODRL is the rights element (see Figure 3.1), which represents one rights expression (e.g. a license, contract, etc.). The rights element can contain the rights expression itself with the party, asset and permissions elements or, alternatively, it can use the offer/agreement element to indicate semantically that a given rights expression is an offer or agreement. ODRL offers three different types of constraints: requirements, constraints, and conditions.

- If a requirement is defined in ODRL, the permission it is related to may not be granted prior to the fulfillment of this requirement. Payments are the most common requirements of ODRL.

- The constraint element in ODRL is designed to narrow ODRL permissions. For example, the permission play can be constrained to five times, by using the bound constraint count. ODRL provides for user-, device- bound-, temporal-, aspect-, target-, and rights constraints.

- An ODRL condition is oppositional to ODRL requirements. Once a condition is fulfilled, the respective permission is revoked.

If the ODRL rights expression includes a digital signature, the corresponding XML Signature [BBF+02] conforming information can be integrated into the document by means of the ODRL signature element. The ODRL syntax allows the addition of XML elements that are compliant with the XML Signature namespace. Figure 3.1 illustrates the elements discussed, which are merely a subset of the ODRL syntax. The entire current foundation model of ODRL is shown in Appendix A 9.1 (for a full

\(^7\)See: http://www.colis.mq.edu.au/

\(^8\)See: http://www.thelearningfederation.edu.au/
description of the concept, please refer to Iannella [Ian02b]). All ODRL elements can be further described by means of an ID, name, etc. with the help of the context element (not shown in Figure 3.1). Note that the ODRL model is further developed on the basis of the data model presented in this thesis. Please find the latest development of the ODRL foundation model at the ODRL Initiative web site⁹.

Excursus: XML as Language Definition Framework

Extensible Markup Language (XML) [BPSMM00] is a meta-language for the definition of application, respectively domain-specific markup languages. Consequently, XML is a means to define new markup languages for a certain domain, e.g. rights expressions. XML is a successor of the Standard Generalized Markup Language (SGML) [ISO86]. A new XML-based markup language defines XML elements and their XML types (e.g. complex type), as well as their structure, i.e. how these elements may be arranged in an instance of the language. XML documents have a well-defined document structure and are human as well as machine readable. XML plays an increasing role in the exchange of data on the web and elsewhere.

The code that defines new markup languages or document types in XML is called Document Type Definition (DTD) [BPSMM00] or XML Schema [TBMM01, BM01]. In contrast to DTDs, XML schemata provide rich datatyping capabilities for elements and attributes (e.g. String, Integer),

⁹See: http://www.odrl.net/
object-oriented design principles (e.g. inheritance), and namespaces. A document that is derived from a specific XML schema or DTD is called XML document or XML instance. An XML instance is valid if it conforms to that specific XML schema or DTD. In order to check whether an XML instance is valid (with respect to a specific XML schema or DTD) is is ‘validated’ against the respective XML schema or DTD (see the Example below). An XML instance is well-formed if it contains at least one element, if it has a unique opening and closing tag, if the tags are nested properly (i.e. there must be an opening and a closing tag that do not overlap), and if the attribute values are quoted [BPSMM00]. XML documents contain one root element; an XML root element and its nested elements below is sometimes referred to as XML tree.

All rights expression languages introduced in this chapter are defined in XML schema documents. For the better understanding of subsequent chapters, a small introduction to XML Schema is exemplified with ODRL. To make the example more comprehensive, the following code (see Figure 3.2) is a simplified subset of the XML Schema defining the Open Digital Rights Language (for the complete ODRL schema, see Appendix A)

As mentioned above, XML Schema documents define permitted elements, as well as their types and structure. In the example, a number of elements are defined, namely rights, offer, agreement, as well as asset, party, permission, and context and finally uid, date, and remark. Each of these elements have types; element types can either be simple or complex. Elements with simple types do not have further elements, and represent the ‘leaves’ of an XML tree, whereas complex element types comprise one or more further elements. The elements right (rightsType), offer, agreement (offerAgreeType), and context have complex element types. An element of the rightType may comprise offer and/or agreement elements.

To reduce complexity, within the definition of a complex type it can be referred to element definitions, e.g. as to offer and agreement in the rightsType definition. The elements offer and agreement are of the complex offerAgreeType, comprising the elements context, party, asset, and permission. Finally, the element context is defined as complex type, comprising the elements uid, date, and remark. The element uid, date, and remark, as well as the elements asset, party, and permission are of the simple type xsd:string and may therefore simply comprise words or sentences.
Figure 3.2: A simplified subset of XML schema defining ODRL
<?xml version="1.0" encoding="UTF-8"?>
<rights xmlns="http://odrl.net/1.1/ODRL-EX"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://odrl.net/1.1/ODRL-EX
    http://wu.wi-wien.ac.at/1er_incl..wir/Guth/schemas/odrl-simple.xsd">
    <agreement>
        <context>
            <uid>agreement #112233</uid>
            <date>12/31/03</date>
            <remark>This agreement was concluded in Vienna/Austria</remark>
        </context>
        <party>Susanne Guth</party>
        <permission>play</permission>
        <asset>Bit clip #999888</asset>
    </agreement>
</rights>

Figure 3.3: A valid language instance of the simplified ODRL schema

At the beginning of the XML schema, there is a definition of which character set can be used in instances (UTF-8), which namespace has been used to write the schema (http://www.w3.org/2001/XMLSchema), and which namespace the schema at hand provides (http://odrl.net/1.1/ODRL-EX). The XML schema also specifies that all elements in instances of this schema have to be assigned to a qualified namespace (elementFormDefault). The attribute attributeFormDefault denotes that only the globally declared attributes must be namespace qualified in instance documents; locally declared attributes are not namespace qualified. From the schema above a respective ODRL instance can be derived (see Figure 3.3). It is important to note that this is a simplified XML instance that is not compliant with ODRL version 1.1. The two XML schemata defining ODRL version 1.1 can be found in Chapter 9 (Appendix A). Stating rights information in an XML-based language provides flexibly, as language elements from other XML schemata can be integrated, i.e. XML-based RELs permit to reuse of description languages, such as the Learning Object Metadata (LOM) standard [IEE02] or Dublin Core [Dub01] or XML Signature [BBF+02].

Several free open-source tools are available for the work with XML schemata and instances. Tools for processing XML documents are called XML parsers.
**XML Parser.** An XML parser, such as Expat\(^{10}\) or Xerces\(^{11}\), is a software tool that receives input in the form of XML markup tags and breaks them up into parts (for example, the nouns, verbs, and their attributes (or options)) that can then be managed by other software services, e.g. a language interpreter. Some parsers include XML validators, such as PyTREX\(^{12}\) (for Python platform) or DSD Processor\(^{13}\) (for Java platform). A validator checks if an XML document is valid in respect of a certain XML schema. A large number of other XML parsers can be found on the W3C web page for XML schema\(^{14}\).

**ODRL Example**

ODRL version 1.1 includes two XML schemata: one that defines the language syntax and a second that defines the ODRL rights data dictionary. The XML schema defining the ODRL rights data dictionary is basically an extension of the XML schema defining the ODRL syntax. For example, in the syntax an element called `permissionElement` is defined. Via the XML mechanism `substitutionGroup` the data dictionary defines all terms that can be used as `permissionElement`, e.g. play, print, copy, etc. Both the XML schema of the ODRL syntax and rights data dictionary can be found in Appendix A. The following code is an example compliant to ODRL version 1.1, showing a contract for a video (disregarding XML namespace labels). ODRL uses XML attributes to assign additional information to the ODRL vocabulary (see “currency” of the amount tag).

The sample license shows a recording of a marketing lecture sold to the *Université Libre de Bruxelles* for the price of €10.00 with the permission to *play* the video *five times*. The video stream’s rights holder is the *Department of Information Systems at the Vienna University Economics and BA*. In this example, ids from the numbering system of the Universal Project\(^{15}\) are used.

```xml
<rights>
  <agreement>
    <party>
```

\(^{10}\) See: http://www.jclark.com/xml/
\(^{11}\) See: http://xml.apache.org/xerces-c/
\(^{12}\) See: http://pytrex.sourceforge.net/
\(^{13}\) See: http://www.brics.dk/DSD/
\(^{14}\) See: http://www.w3.org/XML/Schema
\(^{15}\) See: http://www.ist-universal.org/
This example reflects the ODRL syntax illustrated in Figure 3.1. The basic elements within an agreement are party, asset, and permission. The party element occurs twice, for the consumer and for the rights holder. The rights holder is identified by the <rightsholder> element nested below a party element. The language specification defines that permissions on the same XML tree level as assets, refer to these assets (if no further references are specified). Likewise, assets and permissions are related to customers. The example above does not use a signature element. Constraints (such as ODRL requirements and ODRL constraints) are directly nested below ODRL permissions.
3.4.2 eXtensible rights Markup Language (XrML)

The eXtensible rights markup language (XrML) [Con00] is a rights expression language developed by ContentGuard16, a spin-off of Xerox in cooperation with Microsoft. The language XrML itself is free of charge, but ContentGuard holds a US patent on the usage of rights expression languages in general. ContentGuard claims that its patents pertain "to the distribution of digital works and to any rights language". The interpretation and the consequences of this patent are not clear and are often discussed in DRM/REL-related communities. After working through various online resources, such as web pages and news groups, and personally discussing this issue with ContentGuard employees, I have come to the conclusion that usage rights for industrial/commercial use in the US of XrML or any other rights expression language need to be licensed with ContentGuard.

ContentGuard aims at applying XrML in DRM systems that focus on the commercial exchange of digital goods. Therefore, the standard vocabulary of XrML is designed to express a large number of pricing models. Today XrML is used in Microsoft products (see Section 3.5). Most likely, the further development of XrML will be industry-driven. Is is likely that Microsoft will be able to use its impact to the further development of XrML and the above named patents as competitive advantage in the market for DRM system solutions.

XrML Example

XrML is defined by three XML schemata: the XrML core schema, the XrML standard extension (sx) schema and the XrML content extension (cx) schema. The following example includes XML namespace information, which is necessary for the validation of elements from different namespaces. Just as ODRL, XrML envisages the use of XML Signature [BBF+02] to specify the identity of the contracting parties. The example below shows an XrML instance which reuses elements of the XML Signature namespace.

The "license"-tag is the root element of an XrML instance, resource and party are referred to as the "resource" and "principal" in the basic syntax of XrML. "Grant" includes the actual rights expression. Operations are expressed as "rights" and constraints as "conditions". The XrML-compliant representations of resource and consumer party are "digital work" and "key-

16See: http://www.contentguard.com/
Holder.” The XrML vocabulary contains “print” and “validityInterval” as an operation and condition. The XrML license below grants the owner of the x509 certificate the use of someResource until the end of 2005.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<license xmlns="http://www.xrml.org/schema/2001/11/xrml2core"
    xmlns:digisign="http://www.w3.org/2000/09/xmldsig#"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    ..\schemata\xrml2cx.xsd">
    <grant>
        <keyHolder>
            <info>
                <digisign:x509Data>
                    <digisign:X509IssuerName>CN=Guth Susanne,
OU=Dept. of Information Systems,
O=Vienna University of BA, L=Vienna,
ST=Vienna, C=Austria
</digisign:X509IssuerName>
                    <digisign:X509SerialNumber>12345678</digisign:X509SerialNumber>
                </digisign:x509Data>

                <cx:locator>
                    <nonSecureIndirect URI="http://www.wu-vien.ac.at/someResource"/>
                </cx:locator>
            </info>
        </keyHolder>

        <cx:print/>
        <cx:digitalWork>
            <cx:locator>
                <nonSecureIndirect URI="http://www.wu-vien.ac.at/someResource"/>
            </cx:locator>
        </cx:digitalWork>

        <validityInterval>
            <notAfter>2005-12-24T23:59:59</notAfter>
        </validityInterval>
    </grant>
</license>
```

In XrML the elements keyHolder, operation (e.g. print), digitalWork, and constraint (e.g. validityInterval) are positioned on the same XML tree level.
Therefore, like in ODRL, the keyHolder is related to the operation print and print refers to someResource. Likewise, the constraint is referred to the operation print. The latter semantics is different from ODRL, where constraints are nested directly below the respective operation. Thus, in XrML the constraints are related to the operation and the resource (digitalWork), whereas in ODRL the constraints are related only to the operation. This fact illustrates one of a probably large number of differences in the syntaxes of the two languages. Such syntax differences between the two RELs are hard to identify as unfortunately, until now no formal semantics has been developed, neither for ODRL nor for XrML. XrML seems to be focused more on the commercial aspect of a rights expression language, i.e. on expression licenses that a sold and issued by a DRM platform and bought respectively executed by consumers. In contrast to ODRL, XrML does not seem to focus on the formulation of contracts as in XrML not contract parties are specified but issuers and keyholders.

3.4.3 MPEG 21

The Moving Picture Experts Group (MPEG)\(^\text{17}\) is the ISO/IEC working group in charge of developing standards for the coded representation of digital audio and video. Among other standards, MPEG is working on MPEG 21 with the intention to develop a standardised multimedia framework. Parts 5 [DWW03] and 6 [BR02] of the MPEG 21 standard specify a REL respectively RDD suitable for such a framework. After defining the requirements for RELs and RDDs [Bor02], MPEG issued a call for contributions to select one REL and one RDD as a basis for future development. XrML version 2.0 has been accepted as basis for the development of a future MPEG 21 rights expression language, and the data dictionary from the <indecs>-initiative has been accepted as the basis for the future Part 6 (RDD) of the MPEG 21 standard.

The <indecs>2rdd Project

The <indecs>2rdd project is based on the <indecs> project, which defined a framework for interoperable metadata in content-based e-commerce and is now hosted by the DRM consulting company Rightscom\(^\text{18}\). In contrast to ODRL and XrML, the project does not provide a syntax, but focuses

\(^\text{17}\)See: http://mpeg.telecomitalialab.com/

\(^\text{18}\)See: http://www.rightscom.com/
exclusively on defining a rights data dictionary. Thus, 2rdd is not a rights language but can be adapted from a REL as RDD. The rights data dictionary of the 2rdd project aims at providing a more sophisticated RDD than XrML and ODRL do, and therefore introduces a rights ontology which supports interoperability between the various RELs. The 2rdd project is currently working on the shaping of its RDD according to the requirements of MPEG 21.

3.4.4 LicenseScript

LicenseScript [CCL+03] is a rights expression language that is not defined in an XML schema. LicenseScript is a multi-set rewriting/logic-based language for expressing dynamic conditions of use of digital assets such as music, video or private data. LicenseScript differs from the other DRM languages in that it does not express a certain state, such as an XML contract, which states an agreement at a certain date, but it tracks the development of a rights expression from its issue to its consumption. This REL does not intend to provide a rights expression exchange format between differently designed DRM components or systems. Therefore, each system that aims at using LicenseScript has to use the implementation of the LicenseScript interpreter. Although this characteristic is reducing semantic errors when interpreting rights expressions, it also restricts the usage spectrum of LicenseScript. Each rights expression in LicenseScript is a small Prolog program. Therefore, the question arises if LicenseScript is a rights expression language. Another approach of logic-based rights expression languages has been discussed in [Sza02].

3.5 Current Market Situation and Trends

This section examines the application of rights expression languages in the current DRM systems market. The leading developers of DRM middleware are IBM, Adobe, Real Networks and Microsoft. Real Networks, however, is currently not using any of the introduced RELs in their products.

- IBM has developed a product called the Electronic Media Management System (EMMS)\(^{19}\), which currently deploys a proprietary rights expression language influenced by ODRL. EMMS supports a variety of media formats. IBM is working in close cooperation with Nokia to develop solutions for the mobile communications sector [Nok01]. Nokia

\(^{19}\text{See: http://www.ibm.com/software/data/emms/}\)
has just released a new version of their content publishing toolkit that provides a content creation that meets the requirements of OMA (OMA uses ODRL, see Section 3.4.1) and enables deployment of content and rights to mobile handsets.

- Microsoft has implemented XrML in its Windows Media™ Rights Manager. This software provides a means of packaging content and specifying usage and access rights formulated in XrML. The output of this tool is a file in the Windows Media format (WMA). XrML instances can be interpreted and processed, i.e. enforced, by the Windows Media Player.

- Adobe offers DRM solutions for the exchange of documents including e–Books in PDF. The documents are created with the Adobe Content Server software and can be interpreted and enforced with the corresponding reader, which offers the proprietary functionality of a secure viewer. Adobe is a supporter of the ODRL initiative and a DRM player which will potentially use ODRL in future products. Today, a proprietary format to express rights is used in Adobe’s software.

Based on this middleware, some implementations have already appeared on the Internet. One of the first music subscription services, PressPlay20, uses the Microsoft solution and thus works with XrML. MusicNet21 is a digital music service based on Real Networks’ technology. The M–Stage Mobile Music Service22 is a product on the Japanese mobile–commerce market hosted by NTT DoCoMo, based on IBM’s EMMS technology. Apart from the market leaders, there are also other projects which have implemented rights languages, such as the “Collaborative Online Learning & Information Services (COLIS)23 project, which uses ODRL.

One good source of online information on RELs is the XML coverpages of OASIS’ The XML Coverpages24. Another online source for DRM news (currently free of charge) is DRM Watch25, which has become a commercially run platform for DRM content.

20See: http://www.pressplay.com/
21See: http://www.musicnet.com/
22See: http://www.nttdocomo.co.jp/p_s/mstage/music/
23See: http://www.colis.mq.edu.au/
24See: http://xml.coverpages.com/drm.html