

# Semantic Representation and Enforcement of Electronic Contracts on Audiovisual Content

by Víctor Rodríguez Doncel

Submitted in partial fulfilment of the requirements for  
the degree of Doctor of Philosophy  
Universitat Politècnica de Catalunya

Department d'Arquitectura de Computadors  
Advisor: Dr. Jaime Delgado Mercé

Barcelona, February 2010



# Abstract

*This Thesis presents a computer representation language for agreements on multimedia content and the mechanisms to execute the agreements in the framework of a content distribution system. The presented model is able to represent complex clause terms and can be interpreted by logic reasoners. This work is assumed to be in the wider context of a multimedia distribution system, and it has been integrated with other elements defined in the international standard ISO/IEC 21000.*

This Thesis presents a computer representation language for agreements on multimedia content and the mechanisms to execute them in the framework of a content distribution system.

Content distribution systems trade audiovisual material in exchange of economic compensation in a framework of secure transactions between identified users, what is generally known as Digital Rights Management (DRM) systems. Some DRM platforms have been deployed by private initiatives following closed specifications not intended to interoperate with other systems and not conforming to the standards. Some other initiatives, resulted from the cooperation of corporations and universities in consortia have chosen open design solutions and even open implementations to deploy the content exchange systems. One of the most notable frameworks for the creation, protection, distribution and rendering of multimedia content has been that of MPEG (Moving Pictures Experts Group), whose specification is open, has partial code availability and it has been sanctioned as a collection of International Standards promulgated by the International Organization for Standardization (ISO) with worldwide effect.

Nevertheless, neither the MPEG framework -which is the reference platform taken for this work- nor the other existing DRM systems handle consistently the complete intellectual property value chain and they tend to focus almost exclusively on distributor to consumer transactions. Consequently,

the DRM platforms express the transactions agreements with language elements enough to represent well end user licenses but only these. These digital licenses are written in the so called Rights Expressions Languages (REL), well defined XML documents, from which the MPEG-21 REL is a paradigm and the reference of this Thesis. REL licenses are issued by the right owners of the intellectual property protected material to authorise the execution of a certain action by a certain user, who in exchange has possibly paid an amount of money called fee. These authorised actions result almost always in the rendering operation, which is the most common task for an end user or consumer. However, some intermediate operations are needed to achieve the final product, operations which start with the work conception -from an intellectual property point of view- and continue possibly with adaptations, performances, productions and editions before the final product is broadcasted or distributed. These operations are executed by different actors exchanging also content in economical transactions regulated in narrative contracts and which so far have been somewhat neglected by the existing RELs.

It is a position of this work that the MPEG-21 REL can be used to represent and enforce the agreements of these transactions if the language is properly extended or even substituted with a new semantic representation. For this regard, it is fundamental the analysis of the intellectual property value chain and its most precise description. The intellectual property value chain is described with the Media Value Chain Ontology proposed in the Thesis, a representation in OWL (Ontology Web Language) of the workflow of multimedia content and the successive transfers of intellectual property. Based on the international consensus on intellectual property, the Media Value Chain Ontology describes the different kinds of intellectual property objects (original works, manifestations, instances etc.), the different acknowledged roles intervening in the process (creator, distributor etc.), and the different actions relevant to the intellectual property (both a public communication and a marketing campaign may be relevant in the commercial process but only the first regarding the intellectual property). It also covers the precise relations between these entities, and it provides as well the mechanisms to represent authorisations and bans.

The formalization of the model under the terms of Description Logics allows performing reasoning operations and the implementation of advanced authorisation mechanisms of the licenses expressed with the ontology.

This Thesis proposes the conversion of narrative contracts (contracts written in plain English) in the audiovisual market to the format of digital licenses. This conversion, on despite of the unavoidable loose of informa-

tion, may be of interests if as a result of it, the transaction can take place in the digital environment and governed by the derived digital license. This is achieved by means of a rich knowledge representation schema, where the obligations and permissions exchanged in a contract are dealt as expressions of the deontic logic, and they can determine the behaviour of the distribution platform. A semantic authoriser is proposed which is able to execute rules, evaluate the conditions and the context and give the proper response. To minimize the loose of information, the non enforceable contract clauses are structured into a standard electronic format called *eContracts*. This eContracts format, neutral regarding the contract theme, is enriched with the elements of the Media Value Chain Ontology to provide an adequate representation.

The representation of the value chain proposed here, as well as the conversion from contracts to licenses, has been a part of different projects and has worked tightly integrated with other elements of wider Digital Rights Management systems. In practice, and in order to easily access to the ontology features, an API (Application Programming Interface) in Java is also proposed. On top of it, several applications have been developed. The most paradigmatic acts as a server, and being located in a remote server it is able to receive registration requests and resolve queries intelligently.

Users request their registration in the ontology server, which automatically maps their unique identifier to an instance of the ontology. But users also request the registration of their intellectual property objects, and the permissions they declare, which are also stored in the ontology as new class instances. With all this information and the basis knowledge in the model, the ontology server is able to resolve queries.

These queries serve the different elements in the DRM architecture to validate their intellectual property operations. For example, the ontology server is able to answer which legal nature has a given resource, who is the rights owner of the resource, which is its provenance (the provenance being the intellectual property object from which it derives), which actions are allowed, etc.

The representation of the value chain presented in this Thesis is a complete work but its applications can still be further developed. The authorisation mechanisms can be put in practice with more complex rules, taking advantage of the interoperability of the Semantic Web data.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Aims and objectives . . . . .	1
1.2	Methodology . . . . .	3
1.3	Thesis outline . . . . .	4
<b>I</b>	<b>State of the Art and Legal Framework</b>	<b>5</b>
<b>2</b>	<b>Legal Framework</b>	<b>7</b>
2.0.1	International treaties on Intellectual Property . . . . .	7
2.0.2	Legal Framework on Intellectual Property . . . . .	9
2.0.3	Features of the Intellectual Property Law . . . . .	10
<b>3</b>	<b>Knowledge representation towards reasoning</b>	<b>13</b>
3.1	Logic systems . . . . .	15
3.1.1	Propositional Logic . . . . .	15
3.1.2	First Order Predicate Logic . . . . .	17
3.1.3	Modal Logic . . . . .	21
3.2	Description Logics . . . . .	22
3.2.1	Description Logic semantics . . . . .	23
3.3	The Semantic Web . . . . .	26
3.3.1	The Semantic Web vision . . . . .	26
3.3.2	The Semantic Web technologies . . . . .	28
3.4	Practical use of Ontologies . . . . .	44
<b>4</b>	<b>Contract representation</b>	<b>47</b>
4.1	Digital Representation of Contracts . . . . .	47
4.1.1	Objectives of a contract representation . . . . .	47
4.1.2	A sample audiovisual contract . . . . .	49
4.1.3	Overview of digital representations of contracts . . . . .	52

4.1.4	Contracts Expression Language . . . . .	53
4.1.5	OASIS eContracts . . . . .	55
4.2	Value Chain Representations . . . . .	56
4.2.1	Functional Requirements for Bibliographic Records . . . . .	57
4.2.2	Conceptual Reference Model and others . . . . .	61
<b>5</b>	<b>Rights Expression Languages in DRM systems</b>	<b>63</b>
5.1	DRM Systems . . . . .	63
5.1.1	Overview of some DRM systems . . . . .	64
5.1.2	Common elements in DRM systems . . . . .	71
5.2	MPEG-21 as a Framework for DRM Systems . . . . .	73
5.2.1	Overview of MPEG-21 . . . . .	73
5.2.2	MXM as a MPEG-21 implementation . . . . .	76
5.3	Right Expression Languages . . . . .	81
5.3.1	MPEG 21 REL . . . . .	81
5.3.2	ODRL . . . . .	86
5.3.3	Other REL languages . . . . .	87
5.3.4	Semantic RELs . . . . .	94
<b>II</b>	<b>Contribution</b>	<b>97</b>
<b>6</b>	<b>The Media Value Chain Ontology</b>	<b>99</b>
6.1	Industrial Value Chain and Intellectual Value Chain . . . . .	99
6.2	The Intellectual Property Value Chain Model . . . . .	100
6.2.1	IP entities . . . . .	102
6.2.2	Users . . . . .	104
6.2.3	Actions . . . . .	105
6.2.4	Permissions . . . . .	105
6.3	The Media Value Chain Ontology . . . . .	108
6.3.1	Ontology development . . . . .	108
6.3.2	Namespaces and metadata . . . . .	109
6.3.3	Ontology classes . . . . .	110
6.3.4	Ontology properties . . . . .	115
6.3.5	Ontology restrictions . . . . .	118
6.4	MVCO evolution . . . . .	121
6.5	Alignment and extension mechanisms . . . . .	125
6.5.1	CreativeCommons: An example of MVCO extension . . . . .	126
6.6	Scenario of use . . . . .	128
6.6.1	The MVCO in a Content Distribution System . . . . .	128



6.6.2	Using MVCO: a walkthrough . . . . .	130
6.6.3	Class individuals in several MVCO use cases . . . . .	132
6.7	Example of a rights transfer with MVCO . . . . .	134
6.7.1	An exemplar license . . . . .	135
6.7.2	Equivalent Permission in core MVCO . . . . .	136
6.7.3	Equivalent Permission with extended MVCO . . . . .	136
<b>7</b>	<b>Agreements representation</b>	<b>139</b>
7.1	Analysis of Contracts in the Value Chain . . . . .	139
7.1.1	Parties in audiovisual contracts and the Value Chain . . . . .	141
7.1.2	Rights in audiovisual contracts . . . . .	143
7.1.3	Conditions in audiovisual contracts . . . . .	143
7.2	Using MPEG-21 REL for contracts . . . . .	144
7.2.1	Mapping of contract clauses to license elements . . . . .	146
7.2.2	MPEG-21 REL extension for contracts . . . . .	149
7.3	eContracts representation . . . . .	152
7.3.1	Enforceable clauses with the MVCO permission model . . . . .	154
<b>8</b>	<b>Implementation results</b>	<b>157</b>
8.1	The Ontology in the project AXMEDIS . . . . .	157
8.1.1	Introduction . . . . .	157
8.1.2	Semi-automatic extraction of licenses from contracts . . . . .	159
8.1.3	Synthesis of narrative contracts from licenses . . . . .	161
8.2	The Ontology in the project AX4HOME . . . . .	162
8.3	The Ontology in the E2E project . . . . .	163
8.4	The Ontology in the project VISNET-II . . . . .	165
8.5	The Media Value Chain Ontology APIs . . . . .	168
8.5.1	API methods . . . . .	169
<b>III</b>	<b>Conclusions</b>	<b>173</b>
<b>9</b>	<b>Conclusions</b>	<b>175</b>
9.1	Outreach of this Thesis . . . . .	175
9.2	Contributions . . . . .	176
9.2.1	Refereed publications . . . . .	176
9.2.2	Standardization contributions . . . . .	177
9.3	Future lines of work . . . . .	179



# List of Figures

3.1	Language levels of formality . . . . .	14
3.2	The Tim Berners-Lee's Semantic Web stack . . . . .	27
3.3	The current Semantic Web stack . . . . .	28
3.4	Visual representation of the contract XML Schema . . . . .	32
3.5	RDF triple . . . . .	33
3.6	RDF example . . . . .	34
4.1	Classification of clauses in the CEL model . . . . .	53
4.2	The CEL contract model. . . . .	54
4.3	Top elements in the eContracts XML contract . . . . .	55
4.4	The most simplistic value chain . . . . .	56
4.5	Value chain in an end to end scenario . . . . .	56
4.6	FRBR basic entities . . . . .	58
5.1	OMA DRM Architecture . . . . .	65
5.2	Windows Media DRM Architecture . . . . .	66
5.3	Helix DRM Delivery Platform . . . . .	67
5.4	AXMEDIS Platform . . . . .	68
5.5	OpenSDRM architecture . . . . .	69
5.6	DReaM Architecture . . . . .	70
5.7	MXM Components model . . . . .	78
5.8	MXM Architecture . . . . .	80
5.9	REL license . . . . .	83
6.1	Relations between actions, users and IP entities . . . . .	100
6.2	Main stages in the IP value chain . . . . .	102
6.3	Shortened value chain and accumulation of roles . . . . .	102
6.4	IP Entities transitions in the Value Chain . . . . .	103
6.5	IP Entities in the Value Chain . . . . .	106
6.6	Value chain of IP entities, user roles and actions . . . . .	107

6.7	Permissions in the Value Chain . . . . .	108
6.8	Early Media Value Chain Ontology . . . . .	123
6.9	MVCO Server . . . . .	129
6.10	Three class individuals in the ontology . . . . .	132
6.11	Individuals when Alice makes an Instance . . . . .	133
6.12	Alice's Instance is broadcasted . . . . .	133
6.13	Bob's authorisation . . . . .	134
6.14	Alice's authorisation . . . . .	134
7.1	Contracts and the Value Chain . . . . .	142
7.2	validityPeriodRenew element . . . . .	151
7.3	nonTerritory element . . . . .	151
7.4	interestFee element . . . . .	151
7.5	feeFlatExt element . . . . .	152
7.6	Semantic authorisation . . . . .	155
8.1	Semi-automatic license generation from contracts. . . . .	160
8.2	The Media Value Chain within the AX4HOME project. . . . .	164
8.3	Common Operative Environment (EOC) . . . . .	165
8.4	Adaptations need to be authorized . . . . .	166
8.5	Adaptation authorization in a Virtual Collaboration Scenario . . . . .	167

# List of Tables

3.1	Namespaces used in this document . . . . .	33
3.2	Expressivity of OWL sublanguages in version 1.0 . . . . .	38
3.3	OWL constructs . . . . .	39
3.4	OWL axioms . . . . .	39
3.5	Some semantic web tools . . . . .	46
4.1	Sample contract (I preamble) . . . . .	49
4.2	Sample contract (II, clauses 1-6) . . . . .	50
4.3	Sample contract (III, clauses 7-11) . . . . .	51
5.1	MPEG-21 core rights . . . . .	84
5.2	MPEG-21 multimedia extension rights . . . . .	85
5.3	MPEG-21 rights in the profiles . . . . .	85
5.4	ODRL permissions . . . . .	88
5.5	METS rights . . . . .	90
5.6	TV-Anytime rights . . . . .	91
6.1	MVCO milestones and evolution . . . . .	124
7.1	Different kind of resources traded in analyzed contracts . . . . .	140
7.2	Features of selected sample contracts . . . . .	141
7.3	Different contract categories in the set . . . . .	143
7.4	Rights found in contracts(I) . . . . .	144
7.5	Rights found in contracts (II) . . . . .	145
7.6	Some conditions found in contracts . . . . .	145
7.7	Classification of parties from the rights . . . . .	147
7.8	Mapping of rights from contracts to MPEG-21 REL . . . . .	148
7.9	Clauses classification in eContracts extension . . . . .	152
8.1	Mapping of DIA/CAO/MVCO . . . . .	168



# Listings

3.1	A sample XML contract . . . . .	30
3.2	A sample XML Schema for contracts . . . . .	31
3.3	A sample RDF Schema . . . . .	35
3.4	A sample RDF query . . . . .	36
3.5	A sample SPARQL query . . . . .	36
3.6	OWL fragment in RDF/XML syntax . . . . .	40
3.7	Individual declaration in OWL . . . . .	40
3.8	Sample SWRL rule . . . . .	42
4.1	Sample contract in CEL format . . . . .	54
6.1	Namespace declarations in the MVCO . . . . .	109
6.2	User class and subclasses . . . . .	110
6.3	IP Entity class and its subclasses . . . . .	111
6.4	Action class and its subclasses . . . . .	112
6.5	Defined classes in the MVCO . . . . .	114
6.6	Functional object properties in MVCO . . . . .	115
6.7	Non-functional object properties in MVCO . . . . .	115
6.8	Datatype object properties in MVCO . . . . .	117
6.9	Datatype object properties in MVCO . . . . .	118
6.10	Restrictions over IPEntity in Turtle syntax . . . . .	118
6.11	The first example of REL license . . . . .	135
6.12	Equivalent core MVCO Permission . . . . .	136
6.13	Extension of MVCO to describe a User's RSAKey . . . . .	137
6.14	User class individuals with RSAKey . . . . .	137
6.15	Play as a new Render subclass . . . . .	137
6.16	TemporalContext as a new Fact . . . . .	137
6.17	Condition imposed to the permission . . . . .	137
6.18	MVCO extension to represent a explary REL license . . . . .	138
6.19	MVCO assertions equivalent to the explary REL license . . . . .	138
7.1	Example of condition mapping . . . . .	149
7.2	Contract front with MVCO parties . . . . .	153

7.3 eContracts clause integrated with a MVCO Permission . . . . 153



# Chapter 1

## Introduction

Article 27.

(1) Everyone has the right freely to participate in the cultural life of the community, to enjoy the arts and to share in scientific advancement and its benefits.

(2) Everyone has the right to the protection of the moral and material interests resulting from any scientific, literary or artistic production of which he is the author.

(The *Universal Declaration of Human Rights*, text approved in the General Assembly of the United Nations in Paris, December 1948)

### 1.1 Aims and objectives

The quotation above is the unabridged Article 27 of the Universal Declaration of Human Rights. The work presented in this Thesis assumes this text as fair and strives at making a small contribution towards its implementation in the form of computer content distribution systems.

Of course, there are some who deny the ethic authority of the Universal Declaration, accusing it of ethnocentrism or accusing it of being biased (for example promoting individualism in detriment of collectives, also a fundamental facet of the human nature). Some of them logically defend mind creations as products of the society (rather than the individual's) and in consequence reject any kind of privilege conceded to the work's ultimate creator. This debate, including that one of private property on intellectual creations<sup>1</sup> or private property at all, is left aside, and intellectual property

---

<sup>1</sup>As a reading for this passionate debate, it can be suggested *Free Culture*, the main

rights will be assumed as axioms or pre-conditions for the rest of the Thesis that will not be discussed any further. Actually, almost the only assumption along this text is that *the mere act of creation bestows property*, as in the Universal Declaration of Human Rights, and which hardly can be deemed to be unfair.

The true Revolution we are living, where the Information Age is ousting the Industrial Age has made intellectual creations readily available free of the Kantian a priori constraints of space and time. Information is everywhere at anytime at least in the so called developed countries and so one of the human dreams has come true. But together with the wonderful gift, this is the man's fate, problems come along. A too powerful tool escapes the control of the man who operates it, and the moral limits blur in front of the shining new technologies. Article 27b of Human Rights is nowadays especially difficult to be made observed, and Digital Rights Management systems in despite of their unpopularity, stand as a fair endeavour.

The work of art and in general any intellectual creation has been always reproducible, but it was not until the Industrial Age when the advent of mechanical reproducibility made possible the massive spread of the works. In his brilliant essay in 1936 [2], Walter Benjamin described the changes in the work of art itself produced by this reproducibility, its lost of what he called *aura* and the social consequences of it. After seventy years we face a new turn of screw: if the Industrial Age made possible reproductions without significant quality loss, the Information Age has made possible the reproduction massive, perfect and virtually at no cost. Before the Industrial Age, value had been in the hands of the artist and only in his, during the Industrial Age value was shifted to producers, manufacturers and distributors. After the Industrial Age, value is moving largely towards end users.

Producers and distributors are fighting to defend intellectual property for the benefit of themselves, and only secondarily for the benefit of the creators, but their role is under question. Before the Internet, there was a one-to-all paradigm where one or very few distributors were sources to many receivers of the content (television, conventional press, music distribution, cinemas). In the Internet scenario, the one-to-all paradigm fades in favour of an all-to-all schema, where general Internet users are both creators and end users at a time and most important, without the concurrence of producers and distributors.

Digital Rights Management systems promoted by distributors are fair and will not probably cease to exist, but in parallel, user generated content

---

book of Lawrence Lessig [1] or the Richard Stallman thoughts found in the Internet.

will have to find ways to be protected and remain respectful with the creator. This is the case of the Creative Commons, where user generated content can be easily labelled with a symbol to grant at least a minimal legal protection to the creator. The Creative Commons initiative has read correctly the sign of the times, being intended for plain users to be written and being intended for plain users to be read. But Creative Commons is only a first step, simple and actually not computer-readable. There is not anything else beyond a logo, and information systems are in general unaware of its nature (an Internet browser is not aware of it, for example). Besides this, complex licensing terms cannot be expressed in Creative Commons and the set of symbols may be too short and in any case unable to account for paid transactions.

The development of the Semantic Web in the last years has provided a set of mature tools (but not yet fully exploited) which enable the rich expression of knowledge and the reasoning over it, making systems in the Internet more easily integrable and information better organised than ever.

It is the position of this Thesis that the Semantic Web can be used for a better representation of rights expressions and agreements on intellectual property, and that this representation can be used to drive content distribution systems fair to anybody providing value to the intellectual property objects, in a word, to the different actors of the value chain.

Along this Thesis, an agreement representation formalism will be described, as well as the mechanisms to execute it. Moreover, it will be demonstrated their use and their practical applications.

## 1.2 Methodology

The explosive growth and the novelty of the technologies required to achieve the goals of this Thesis present a complex panorama, sometimes confusing, which sees very fast the born and death of promising ideas. Extra effort is needed to discern between ephemeral initiatives and successful ones.

However, the technologist's work alone in this area is condemned to fail if it does not take into account other factors, of legal and social nature. There are many interests and views around the topics dealt in this work, including but not only the view of authors, the interests of producers, distributors and service providers or the preferences of consumers, being subject all of them to the legal provisions. To this regard, along the work to produce this Thesis, there has always been a tight contact with as many as possible of these parties.

In behalf of artists and creators, Marc Gauvin, a representative of the Spanish Author's Collecting Society, has tracked a large part of this work. Also, there has been the formal consultancy provided by the Associazione Fonografici Italiana (AFI) regarding the different kinds of audiovisual contracts and the common practices in the industry. And lastly, it has to be remarked the precious informal contact held in meetings with voices from broadcasters (BBC and European Broadcasters Union) service providers (Telefonica) and free software advocates. Many people are interested in the debate around intellectual property in the Internet time, but these authoritative voices have proved helpful to provide a global view.

Finally, the implementation of the agreements expression systems and its execution has been a definitive feasibility test of the ideas here proposed.

Ontology development is not usually a single event rather a process, and in occasions a fairly complicated process. Because of this, an ontology methodology development has been followed (see Section 6.3.1 for more details) and a track has been kept of all the changes, versions, demos and accompanying software with its release date and version number.

### 1.3 Thesis outline

Part I of the Thesis describes the state of the art, comprising a Chapter for general knowledge representation systems (Chapter 3), a Chapter for specific contract representation formats (Chapter 4), and a Chapter for the license standards used in the different Digital Rights Management systems (Chapter 5). Additionally, Chapter 2 gives an overview of the legal framework of the intellectual property.

Part II of the Thesis is the contribution of this author, structured in the central work, the Media Value Chain Ontology (Chapter 6), its use for agreements representation (Chapter 7) and the implementation results (Chapter 8).

Finally, some final thoughts are developed in the conclusions (Chapter 9), which include also a first glance at the impact of this work and the future lines of research derived from it.

## Part I

# State of the Art and Legal Framework



## Chapter 2

# Legal Framework

The first quotation in the Introduction of this Thesis refers to the Article 27 of the Universal Declaration of Human Rights. However, in order to give a precise representation of contracts on multimedia content, it is necessary to acquire a deeper legal background. This background is provided in this Chapter.

By definition, contracts belong to the sphere of private law, but they are subordinated to the public law limitations. To give an example, a contract may say an author must refuse creating any artistic work for a certain period, but it is needed to know that this contract would not be valid as this clause is declared illegal according to the law.

This Section describes briefly the required legal concepts to continue with the speech, reviewing the international agreements on intellectual property and their implementation in the legal systems. Also, this Section will be useful to acquire familiarity with the particular jargon found in audiovisual contracts which needs to be understood to correctly interpret the contracts.

### 2.0.1 International treaties on Intellectual Property

International agreements and treaties declare a framework of directives whose signing parties (usually sovereign states) have to translate into concrete laws specific for each country. International treaties have also moral authority conferred by the fact that usually the number of signing countries is a vast majority. Additionally their simplicity and general applicability match the purposes of this work. In the next paragraphs the main agreements reached in the area of intellectual property will be enumerated; a deep review can be found in [3].

Before any international agreement, there existed concepts such as *droit d'auteur* (author's right) and *copyright* in different legal systems. Derived from different legal traditions, the British concept of copyright is born from a reaction to printers monopolies at the beginning of the eighteenth century while the French (and continental) concept of *droits d'auteur* aimed at protecting the creator of artistic works. The first view emphasized the economical point of view, while the second the moral. The early international agreements and conventions on intellectual property took ideas from both systems and nowadays their legal prescriptions are merged.

The first important agreement on intellectual property was the the *Berne Convention for the Protection of Literary and Artistic Works*, usually known as *the Berne Convention*, held in Berne, Switzerland, in 1886. Three years before, a similar agreement had been signed in Paris for the protection of patents, trademarks and industrial designs; in Berne the Convention spoke about novels, poems, plays, songs, operas, musicals, paintings, sculptures or even architectural works.

The Berne Convention establishes that rights are given over a work to his author automatically as soon as it has been fixated, without the need for a formal registration. The author entitles all copyrights in the work and to any derivative work, unless the author explicitly disclaims them or the copyright expires. The biggest accomplishment in that time was that the protection took place globally, across the national boundaries, instead of having a protection limited to one country. Since then, the Berne Convention agreement has been amended and modified, and legally it is in force and signed by 164 countries (only missing Iran, Laos and a few others). After the Berne Convention, other agreements followed, protecting not only the authors but also the producers, the distributors and the broadcasters.

As new technologies appeared in the XXth century new provisions had to be made to answer to the new reality. Devices like tape recorders made the reproduction of sounds and images easier and cheaper than ever before, and even the ideas object of protection themselves could be presented in different ways. The Rome Convention came to give protection to performers and producers. In the *Rome Convention for the Protection of Performers, Producers of Phonograms and Broadcasting Organizations*, performers were protected from being recorded or broadcasted during their live performance; broadcasters could now prohibit the re-broadcasting of their broadcasting etc. The rights of the producers were reinforced with the *Geneva Convention for the Protection of Producers of Phonograms against unauthorized Duplication of their Phonograms*, in 1971.

The Berne Convention set up an International Bureau in 1886 (*Bureaux*



*internationaux réunis pour la protection de la propriété intellectuelle*) that with the time was succeeded by the WIPO, World Intellectual Property Organization. WIPO became in 1974 a formal agency of the United Nations and nowadays it manages 24 international agreements, whose content is well summarized in [4].

The main agreements in force are the successors of the first conventions, updated in 1996: the *WIPO Copyright Treaty (WCT)*, updating the Berne Convention to protect authors and the *WIPO Performance and Phonogram Treaty (WPPT)* superseding the Rome Convention and the Geneva Convention to protect performers, producers, broadcasters and distributors rights [5].

After 1996 it took six additional years until WIPO treaties were signed by 30 member countries, the minimum number that the UN considers in order to make its application effective. Nowadays these agreements are universally in force.

## 2.0.2 Legal Framework on Intellectual Property

WCT was implemented in the US in the 1998 through the important law known as Digital Millennium Copyright Act (DMCA [6]). DMCA criminalizes the production and dissemination of technology, devices, or services that are used to circumvent DRM measures, and bans the mere act of circumventing itself, even when there is no infringement of copyright itself. A detailed account of the US legal framework on intellectual property is found in [7].

Three years after, WIPO treaties were also introduced in Europe by the European Union Copyright Directive<sup>1</sup> (2001/29/CE), which was developed in two additional directives. According to these new policies, the EU member states adapted their national laws.

For the particular case of Spain, the law regulating intellectual property is the *Ley de Propiedad Intelectual* or LPI of 11th November 1987 (updating a law of 1879). The LPI was amended in 1992 (to add a canon in analogue devices to compensate the right of private copy) and rewritten in 1996. In 2006 it was again changed (Ley 23/2006), to include the new particularities of the Internet world, and the principles from the EU directive (2001/29/CE) (thus incorporating the ideas of the WIPO agreements aforementioned).

---

<sup>1</sup>Directive 2001/29/EC of the European Parliament and of the Council on the harmonization of certain aspects of copyright and related rights in the information society, 22 May 2001, European Parliament and European Council

Also related important laws in Spain are the *Ley de Internet* and the *Ley de Servicios de la Sociedad de la Información y Comercio Electrónico* (LSSICE), a law regulating electronic commerce and that is based in the European Directive (2000/31/CE). For the interest of this Thesis, it is enough to know that it enables the existence of electronic commerce by acknowledging electronic contracts as full valid contracts (Art. 23).

### 2.0.3 Features of the Intellectual Property Law

#### The object of protection

Berne Convention establishes the protected works as the ‘literary and artistic works’, particularly: *books, pamphlets and other writings; lectures, addresses, sermons and other works of the same nature; dramatic or dramatic-musical works; choreographic works and entertainments in dumb show; musical compositions with or without words; cinematographic works to which are assimilated works expressed by a process analogous to cinematography; works of drawing, painting, architecture, sculpture, engraving and lithography; photographic works to which are assimilated works expressed by a process analogous to photography; works of applied art; illustrations, maps, plans, sketches and three-dimensional works relative to geography, topography, architecture or science..* Also derived works are considered work, like translations, musical arrangements etc., without undermining the original author’s rights.

#### The author

Authors are considered as such as soon as the work has been fixed (i.e. taken a material form). Nevertheless there are databases where works can be registered for an increased protection (e.g. *Registro de la Propiedad Intelectual*). Making a registry confers no additional rights but helps in case of lawsuit. Also, parallel to official organisms, other registration platforms are appearing with increasing acceptance, like Coloriuris<sup>2</sup> or SafeCreative<sup>3</sup> in Spain.

Authors can be individuals or collectives, and there must be consent of all authors to exploit the work. It is also expected from the co-authors to have reached an agreement on the distribution of the benefits from the exploitation rights. For the case of audiovisual works (like a movie), the

---

<sup>2</sup>Coloriuris. Autogestión de derechos de autor y registro de obras. <http://www.coloriuris.net>

<sup>3</sup>Safe Creative. Registro de propiedad intelectual. <http://www.safecreative.org>

authors are considered to be the director, the scriptwriter and the music composer.

### **Author rights**

Author rights are classified as *moral rights* and *exploitation rights*.

Moral rights entitle the author to decide whether the work is going to be published or not, the way it is going to be published (e.g. anonymously), the right to have the work respected in its integrity and in any case the right to be recognised as the author. Moral rights are only transmissible by the author's death to the heirs.

Exploitation rights are those rights whose exercise yield economical profit, and they are essentially *reproduction*, *distribution*, *public communication* and *transformation*. Reproduction means any act of making the work perceivable for the human sense. Distribution includes selling physical items, but also renting, lending etc. Public communication is any act where a plurality of persons have access to the work, excluding the domestic case; being examples the exhibition of a movie, music broadcasting or public exhibitions. Transformations include translations, adaptations and any act which yields a derived work.

Author rights last for during the author's life and an extended period, which for the Spanish case is of 70 years.

The execution of rights is subject to certain exceptions. For example, no author authorisation is needed to reproduce a work if it is going to be done for the sake of public security, or in an official state act, or for education purposes, or as quotation etc.

One of these exceptional situations, and origin of dull polemics is the right to private copy. The Spanish LPI, Art. 31, recognizes the right to private copy, i.e. to make private copies without the explicit permission of the author as long as it is a non for profit act made from an original copy. In exchange, recording devices and storing devices are subject to a special levy to compensate the authors. Brought in income is managed by Author Collecting Societies like SGAE (Sociedad General de Autores y Editores) or CEDRO (Centro Español de Derechos Reprográficos).

### **Transfer of rights**

Unlike moral rights, exploitation rights can be transferred in life to others through an agreement (here it lies the foundations of this Thesis). The transfer may or may not be in exclusive, and unless exceptional situation,

it will entitle the author to have a proportional participation in the benefit (this amount is also known as *royalties*).

The transfer of the exploitation rights from the author to third parties is formalized in contracts. In the *edition contract*, the agreement is to exploit the distribution, and in the case of *musical edition contract*, the right transferred is not only the distribution but also the public communication right. In the *execution contract*, the agreement is to publicly represent a theater play, publicly execute a music work etc. *Production contracts of audiovisual works* entitle the producer the right to reproduce, distribute, publicly communicate, translate and dub the audiovisual work.

Performers and executor artists need the permission of the author to make a performance (a public communication) but in turn they hold the exclusive right to authorize the fixation of their performances (to record them etc.). The reproduction, distribution or public communication of their performances needs to be authorized in written contracts, and additionally it needs the authorisation of the original work's author.

Performers may transfer the right to fixate their performance to *producers of phonograms* or to *producers of audiovisual recordings*, who after having signed a contract are the rightsholders of the fixation and may authorize subsequent reproductions, distributions and public communications.

Broadcasting entities, once acquired the proper rights, may broadcast, authorize the fixation of their broadcasting or their subsequent distribution.

All these rights transfers take the form of contracts which are object of analysis in this Thesis, as well as their implied relations.

## Chapter 3

# Knowledge representation towards reasoning

This Chapter describes the state of the art on formal systems able to represent knowledge and reason on it, with a special focus on the Description Logics branch, which has been the base for this work.

Knowledge representation, in the computer science jargon, deals with how to structure information in a computer system so that it can be easily accessed. The closest language to the human being is the language we speak, also known as *natural language*, extraordinarily flexible and with a lax, vague syntax which often can be broken without loss of information in the message. A text file containing a written contract (also referred as narrative) can be in a digital file but nevertheless it remains a natural language contract.

An attempt to organize the information logically is using a structuring schema, like XML, which offers very precise rules to assess the syntactic validity of a document. XML structures like the eContracts standard (to be reviewed in next chapters) are a first organization of the information in contracts.

If the well formed expressions of a syntactically valid document are also conformant to the rules of a formal system (they can be derived from axioms or derived from formulas achieved after following the formation rules), then the document has reached a semantic validity superior to the syntactic level. At this stage, when combined with the principles of logic, information can not only be obtained as it was introduced, but also in more refined forms. This includes retrieving information which was not explicitly injected, and

Contract as a set of assertions in a formal system (OWL ontology)	Semantically valid document
Contract as a structured document (eContracts XML)	Syntactically valid document
Contract as a narrative document (TXT file)	Natural language

Figure 3.1: Language levels of formality

which is the result of a certain formal reasoning, e.g. inferencing. These three levels are represented in Fig. 3.1.

Ontologies are formal systems able to represent information and suitable for formal reasoning, and have been used in many fields like medical diseases classification, electronic commerce [8] or natural language processing [9] among many others.

In this Thesis, ontologies will be used to describe agreements in the most precise way. The deductive apparatus (axioms and rules) will be needed to perform inferences and handle the contracts with a certain degree of intelligence. An OWL ontology (to be described in this Chapter) will adopt the syntactic form of a XML file (although not necessarily) or even it could be stored as a text file, but the abstract information it conveys is the one at the highest level in the Fig. 3.1.

Formal language and the deductive apparatus form together a system usually called *formal system*. Formal systems exhibit three properties, which may or may not be true:

**Consistence** A calculus is consistent when it is not possible to derive contradictions. All the derivable formulas from the calculus are logical truths.

**Completeness** A calculus is complete when on it is possible to prove all the true formulas doable with symbols.

**Decidability** A calculus is decidable when there exists in it at least a procedure to decide in a finite number of steps whether a formula is true or not.

Needless to say, good systems are those consistent, complete and decidable. Unfortunately, the more expressive the formal language is, the more difficult is to have these good systems. Propositional logic is consistent, complete and decidable but it is not much of help to represent a contract.

Throughout this Chapter, an overview of logic systems will be made before describing in detail the chosen system for this Thesis, the descriptions logics formal system.

### 3.1 Logic systems

A good review of classical logic through the centuries can be found in [10] while a simple introduction covering from propositional logic and first order logic to modal logics in [11]. From a modern point of view, logic is seen in terms of formal systems, which are also more convenient for the computer processing. The author of this Thesis believes that the bibliography is however limited when describing the logic foundations of knowledge representation systems like OWL, existing a gap between the literature written by logicians and that given by computer scientists. This Section gives a succinct introduction to the concepts needed to be understood the ontologies presented in further chapters.

#### 3.1.1 Propositional Logic

Propositional logic is the logic which deals with propositions (assertions which can be either true or false), how they are combined (with logical connectives) and how theorems can be proved. Examples of propositions are:

Ars Magna is authored by Gottfried Leibniz

or:

Ars Magna is authored by Ramon Llull

Atomic propositions are represented with lowercase letters ( $p, q, \dots$  etc.), because it is easier to say that  $p$  is false and  $q$  is true (if  $p$  and  $q$  represent respectively the two previous propositions).

Propositional logic is not really expressive; in the previous example nothing prevents  $p$  and  $q$  from being true at the same time (we have the implicit knowledge that that book is authored by a single person), but propositional logic introduces the basic elements to be used in more complex systems.

The logical connectives can be defined by the truth tables, and they are represented by the well-known symbols: negation ( $\neg$ ), conjunction ( $\wedge$ ), disjunction ( $\vee$ ), conditional ( $\rightarrow$ ), biconditional ( $\leftrightarrow$ ) etc. Negation is said to be a *monadic* operator, as it only operates on a single proposition, and the others are called *diadic* given that they operate on two propositions.

Each expression is either *valid* (always true regardless the values of the atomic propositions) or *contradictory* (always not valid) or *consistent* (valid

depending on the values of the atomic propositions). Valid expressions are also referred as *tautologies*. With the help of the truth tables, each expression can be classified into one of these categories. For example, the expression:

$$z = q \vee \neg q$$

is a tautology, because regardless the truth value of  $q$ ,  $z$  is always true; Ars Magna is either authored by Ramon Llull or not. However, the expression

$$z = q \wedge \neg q$$

is contradictory because whatever happens,  $z$  will be always false (nothing can be true and false at the same time, at least in Aristotle logic). The expression:

$$z = p \wedge q$$

is a consistent expression:  $z$  can be true, or can be false depending on the values of  $p$  and  $q$ . Perhaps the authors were Ramon Llull and Leibniz, perhaps not.

The expressions resulting from combining propositions with connectives (and parenthesis) creates new propositions. The use of capital letters in the expressions is reserved for the variables of variables or metavariables. Thus,  $X \wedge Y$ , represents any two expressions binded by the operator  $\wedge$ , like

$$p \wedge q$$

or

$$p \wedge (r \vee s)$$

etc. Valid inference schemes can be called *laws* of logic, and although there are an infinite number of laws, some of them are named. For example,

$$\neg\neg p \rightarrow p$$

is known as the *Law of Double Negation*, or

$$\neg(p \wedge q) \rightarrow (\neg p \vee \neg q)$$

is known as the *De Morgan Law*.

Propositional logic is of no direct interest for the regards of this Thesis, excepting that it lies the foundations of more complex systems. For example, if we state that

Every author of Thesis is its copyrights holder

and we state that

Victor is the author if this Thesis

propositional logic cannot derive that Victor is the copyright holder of this Thesis. This can be achieved however by the use of *first order predicate logic*.



### 3.1.2 First Order Predicate Logic

Predicate logic is the next step in logic, and it not only explores the truthness of whole propositions but it also provides an insight on its composition. Information such as 'Every P is...', 'Exists at least one P such...' is also handled. Thus, it is also possible to express:

[This document] is a [thesis dissertation]

and also:

Every [thesis dissertation] is [in English language]

allowing us to infer that:

[This document] is [in English language]

This reasoning can be applied regardless the content within the brackets. Expressions now designate *properties* of *individuals* or *relations* between themselves, and the logic system is aware about it. By individual here we understand everything that is subject to receive a proper name. Properties (attributes of an individual) and relations (bindings between two individuals) are called *predicates*. Sentences in predicate logic are thus built up from constants, predicates and functions.

- Constants, referring to objects, like 'Victor', or 'this Thesis'
- Predicates, expressing relationships like 'is a', or 'the rightsholder of'
- Functions, to refer indirectly to other objects, etc.

Monadic predicates are those expressions containing one single individual, while dyadic, triadic and in general polyadic expressions deal with more individuals. Predicates are also functors, much as the connectives in the previous subsection were also functors.

Examples of atomic expression are:

Copyright('this Thesis', 'Victor')

Language('this Thesis', 'English')

to state that 'this Thesis' is written in English language and that the copyright owner of 'this Thesis' is 'Victor'. As with propositional logic, compound sentences can be made with the connectives  $\neg$ ,  $\vee$ ,  $\rightarrow$  etc. (for example, it can be said that 'this Thesis is authored by Victor and it is in English').

Note that for the regards of representing knowledge, there are many alternatives of expression, and it is a task of the modeler to choose the right one. For example, instead of creating a dyadic predicate 'Language', it could have been defined a monadic predicate 'English'.

This is starting to become interesting for our regards, given that we can now assert who is the author of the Thesis or who has the publishing rights of it. For those accustomed to program computers with object oriented languages, the achievements may look louse. Propositional logic is implemented in something as simple as the logic operators of C++ or Java which act on boolean variables. Predicate logic is implemented from the moment that some variables may point to other objects, and when class individuals can be addressed with pointers.

But predicate logic introduces also the quantifier symbols: the existential ( $\exists$  exists at least one) and the universal ( $\forall$  for all), for which there is no equivalent in that C++ or Java. Now more expressive sentences can be represented, like 'Exists an author for each Thesis'.

$$\exists a.(\forall t.Author(t, a))$$

Where  $a$  represents an author, and  $t$  represents a Thesis. Note that these assertions can serve to describe an abstract model of the intellectual property (and the sentence before could be a good example) but also to describe a real situation where authors have names and Thesis titles.

The *first* in First Order Logic refers to the prohibition of referring to the expressions themselves in the expressions. The simplest example creating problems is:

$$[This\ expression] \text{ is } [false]$$

It is easy to identify the contradiction in the self-referring expression above. Higher orders of logics are of no interest for the purpose of this Thesis because these kinds of situations do not happen in ordinary agreements, and in the cases where contract clauses refer to other clauses, it is usually possible to rewrite it easily so that it remains within the limits of first order logic.

First order predicate calculus with only monadic operators is consistent, complete and decidable; but if dyadic operators are introduced the calculus is no longer decidable, as Church showed solving the halting problem [12]. These situations, however, rarely appear in the logic formulas arisen from the agreements statements, and for practical regards we can consider these problems as decidable (any query posed as a formula can be tested and its truth value determined). The calculus of higher order logics poses more serious problems being incomplete, but it shall not be needed here and it can be dismissed.

### Formalisation

An excellent text for the formalization of First Order Logic is [13]. First-order logic can be expressed as a formal language, in which it can be me-

chanically determined whether a given expression is legal (in opposition to natural languages). Expressions are made by: *terms*, representing objects, and *formulas*, which represent predicates that can be true or false. The alphabet is limited to:

- Variables:  $a, b, c, \dots$
- The logical connectives
- The quantifier symbols
- Parenthesis, brackets etc.
- The equality symbol ( $=$ )
- Predicate symbols ( $E(x)$  meaning 'x is English')
- Function symbols ( $f(x)$  meaning 'the author of 'x'')

In formal systems, transformation rules can be applied to create new well-formed formulas, which in turn can suffer new rules and so on in an unlimited process. Standard first order predicate logic defines the following rules:

- R1 A *term* is a variable or a function of variables.
- R2 Predicate symbols. If  $E$  is an  $n$ -ary predicate symbol and  $x_1, \dots, x_n$  are terms then  $E(x_1, \dots, x_n)$  is a formula.
- R3 Equality. If  $x_1$  and  $x_2$  are terms, then  $x_1 = x_2$  is a formula.
- R4 Negation. If  $\phi$  is a formula, then  $\neg\phi$  is a formula.
- R5 Binary connectives. If  $\phi$  and  $\psi$  are formulas, then  $\phi \vee \psi, \phi \wedge \phi$ , etc. are formulas.
- R6 Quantifiers. If  $\phi$  is a formula and  $x$  is a variable, then  $\forall x\phi$  and  $\exists x\phi$  are formulas.

### Interpretations

An *interpretation* is the denotation given to predicate symbols and function symbols. While the other symbols have universal applicability (that is to say,  $\forall$  is always going to be interpreted by everybody in the same way), the predicate and the function symbols are defined each time differently, and

$E(x)$  may mean 'x is in English' in one context but also 'x is an Expert' in other. When modeling knowledge with a first order logic, an interpretation has to be defined.

A *domain of discourse* is a nonempty set of 'objects' of some kind (e.g. Thesis dissertations in a certain department). Predicate symbols are interpreted as tuples of elements in the domain of discourse (e.g. Thesis written in English correspond to a very defined set). Functions symbols are interpreted as functions (whichever how they are defined, like 'number of pages in the Thesis'). The work presented in this Thesis describes a domain of discourse, too, given in this case in English language.

The key idea for this chapter is that given an interpretation, a predicate symbol and a set of elements of the domain of discourse, it is possible to determine whether the predicate is true or not.

If a predicate  $\phi$  evaluates to true under a given interpretation  $M$ , then it is said that  $M$  *satisfies*  $\phi$  (in symbols, it is said that  $M \models \phi$ ). A collection of predicates is *satisfiable* if there is some interpretation under which they are true. Valid formulas are those which are always true regardless the interpretation (they are much like tautologies in propositional logic).

Deductive system are used to demonstrate, on a purely syntactic basis, that one formula is a logical consequence of another formula (i.e. every interpretation that makes true the first formula, makes also true the second). There are different deductive systems, like natural deduction, tableaux algorithms (see [13]) etc. It is very important to note that deductive systems are purely syntactic, and derivations can be verified without considering any interpretation, what is very appropriate for computer processing. Deductive systems can be programmed into computer applications like Prolog.

### **From theory to practice: Prolog**

There have been some formal attempts to standardize the representation of formal expressions, particularly useful for computer exchange. This is the case of Common Logic, a logic framework intended for information exchange and transmission, much in the line of conceptual graphs. Common Logic is an International Standard [14], however it does not propose any idea regarding subsumption or information retrieval.

The way of specifying an interpretation is through structures or *models*. If we wanted to use first order logic to represent the information in a contract, we should give a model. Going from theory to practice, computer programs can be used to specify this model. The previous subsections might have seemed written by a mathematician or a logician, but for the engineer point

of view they have a direct mapping into practical computer applications and programming paradigms. This is the case of Prolog, Lisp or others.

Prolog clauses can be directly translated into first order predicate logic, except for the so-called 'extra logical' and 'meta-logical'. Changing a few symbols, Prolog expressions can be read as good as first order logic expressions (for example, ':-' can be read as  $\leftarrow$ , ',' has to be read as  $\vee$  etc.). For example, the following expression is valid:

$$\text{speaks}(X,L) \text{ :- author}(T,X), \text{ language}(T,L)$$

and can be read that 'if the author of T is X, and T is in the language L, then X speaks L'.

In logic programming systems, like Prolog, knowledge is asserted first (as relationships among data values) and then queries can be launched to see whether certain relationships hold. Prolog is a declarative<sup>1</sup> language which permits expressing facts (like 'Victor is the author of this Thesis' and 'this Thesis is in English') and rules like the one above. Further queries like: 'does Victor speak English?' can be answered later on and reveal Prolog's utility. As will be shown later, some expression languages have used Prolog and Lisp to describe contracts, as it is the objective of this Thesis.

### 3.1.3 Modal Logic

Modal logic is born with Aristotle, but a modern approach can be found in [15]. In the logic we have seen so far, propositions are either true or false. Modal logic deals with modalities of this truth, or nuances in the truth value. Modalities of truth include necessity, possibility, contingency etc.

Modal logic introduces the use of unary modal operators. When applied to the notions of *necessary* (using the symbol  $\sharp$ ) and *possible* (using the symbol  $\diamond$ ), they can be related as follows:

$$\diamond p \leftrightarrow \neg \sharp \neg p \qquad \sharp p \leftrightarrow \neg \diamond \neg p$$

The first expression can be read P is possible if and only if not P is not necessary, and the second as P is necessary if and only if not P is not possible.

Other modal logics can be expressed, for example the temporal logic ( $\sharp$  P meaning that P holds at every moment and  $\diamond$ P meaning that P holds at a

---

<sup>1</sup>Note that conventional programming languages have both declarative expressions (like  $LETA = 5$ ) where no instructions have to be executed and imperative expressions (like  $A = B + C$ ) where the processor has to do something. Prolog is a declarative language because all the expressions (facts and rules) are declarative, being the executive stage after, when queries are launched.

given moment), epistemic logic (about knowledge) or deontic (about duties and obligations). The last one is of special interest for our purposes.

### Deontic Logic

Deontic logic is a formal system that attempts to capture the essential logical features of obligation, permission and related concepts.

If  $\sharp P$  is said to mean ‘it is obligatory P’ and  $\diamond P$  is said to be ‘it is permissible P’, then a formal system can be axiomatized by adding to the classical propositional logic these two axioms (work provided by Kripke [16]):

$$\sharp(A \rightarrow B) \rightarrow (\sharp A \rightarrow \sharp B) \qquad \sharp A \rightarrow \neg(\sharp \neg A)$$

That is to say, for the first we can read *if it ought to be that A implies B, then if it ought to be that A, it ought to be that B*, while for the second we can read *if it ought to be that A, then it is permissible that (i.e. not obligatory that not) A*. While the first and decisive essay on deontic logic was written in 1959, an easy formalisation is given in [17]. For an exhaustive survey, it can be read [18].

Deontic logic, however, is prone to wrong interpretations and paradoxes appear if not carefully avoided. A review of the possible problems is given in [19]. Reasoners with the specific aim of solving deontic questions have been developed [20] and even applied to contracts [21] or other legal cases [22].

## 3.2 Description Logics

In the 1970’s, knowledge representation approaches fell into two categories: logic-based formalisms, which evolved out of the intuition that predicate calculus could be used to represent portions of the world, and other non-logic based representations where structures and relationships were defined *ad hoc* for each problem. The first paradigm gave rise eventually to Prolog, Lisp and other AI programs. The second took the form of *semantic networks* and rule-based representations, and finally derived in Description Logics (DL). The key reference for Descriptions Logics theory is [23].

A semantic network [24] is a graphic notation for representing knowledge in patterns of interconnected nodes and arcs, supporting automated systems for reasoning about them. Nodes represent concepts, usually nouns. Arcs can be usually described with verbs, and they represent relations like subtype, instance, part-whole etc. and thus one node connected to other node by an arc may mean ‘A is a subtype of B’, or ‘A is a part of B’, or ‘A is an instance of B’. Semantic networks also add simple properties to the

concepts, which do not link to other nodes but associate a value to a node: they are known as *attributes*. Semantic networks can be represented with a subset of the first order predicate logic <sup>2</sup>.

Applying transitivity and some other rules to some of these relations, knowledge can be inferred for non directly connected nodes. For example, the relation 'is-a' generates *inheritance networks* where attributes and properties are inherited in the classes from their more general classes. If one attribute of 'books' is 'the number of pages', and 'a Thesis is-a book', then 'a Thesis' has also as attribute 'the number of pages'.

Description Logics is the result of extending semantic networks for an enhanced expressivity while keeping a formal tractability. To give this precise characterization of the meaning of the network, a language for the elements of the structure has to be given, as well as an interpretation of the strings of the language.

### 3.2.1 Description Logic semantics

The Description Logics language has an alphabet of unary predicates (*concept names* or classes), binary predicates (*role names* or properties) and individuals (constants). Classes are unary predicates that are interpreted as sets of individuals while properties are binary predicates that are interpreted as sets of pairs of individuals. Individuals are exemplars or instances of the classes.

Atomic concepts and atomic roles can be linked with a set of constructors to form more complex expressions (concept descriptions). There are different 'Description Languages', and they are distinguished by the constructors they provide. The language  $\mathcal{AL}^3$  (attributive language) is the minimal language of practical interest. If  $A, B$  represent atomic concepts,  $C$  represents a concept description and  $R$  represents a role,  $\mathcal{AL}$  offers the following elements:

- Atomic concept.  $A$ . It means a class.
- Negation.  $\neg A$  All the objects not belonging to  $A$ .
- Universal concept.  $\top$  It means all the objects.
- Bottom concept.  $\perp$  It means none of the objects.

---

<sup>2</sup>Note that semantic networks are not much different from Porphyrian trees, described in the antiquity, or its derived forms in the Middle Age found in Ramon Llull's *Arbor scientiae* and others.)

<sup>3</sup>The expressive power of logic languages is usually described with calligraphic letters.

- Intersection.  $A \sqcap B$  It means the objects belonging to  $A$  and to  $B$ .
- For all.  $\forall R.C$  All the objects who are related through  $R$  to objects only of the class  $C$ .
- Exists.  $\exists R.\top$  All the objects who are related with at least one object through  $R$ .

Note that in  $\mathcal{AL}$ , negation can be applied only to atomic concepts (not to concept descriptions), and existence can be applied only with the universal concept. To give an example of expressivity with these constructs, we can define a class  $P$  of persons, a class  $B$  of books, a class  $J$  of journals and a relation  $W$  meaning 'writes'. The concept description of 'Writer of books' cannot be expressed:

$$\exists W.B$$

because is not a legal expression in  $\mathcal{AL}$ .

An interpretation, as it was described in page 19, assigns real life ideas to atomic concepts and atomic roles. Thus, an *interpretation*  $I$  consists of a domain of interpretation  $\delta^I$  and the function that relates to each atomic concept  $A$  to a subset of  $\Delta^I$  ( $A^I \subseteq \Delta^I$ ) and relates an atomic role  $R$  to pairs of elements of  $\Delta^I$  ( $R^I \subseteq \Delta^I \times \Delta^I$ ).

The construction of concept description can therefore be described in terms of sets:

$$\begin{aligned} \top^I &= \Delta^I \\ \perp^I &= \emptyset \\ (\neg A)^I &= \Delta^I \setminus A^I \\ (C \sqcap D)^I &= C^I \cap D^I \\ (\forall R.C)^I &= \{a \in \Delta^I \mid \forall b.(a, b) \in R^I \rightarrow b \in C^I\} \\ (\exists R.\top)^I &= \{a \in \Delta^I \mid \exists b.(a, b) \in R^I\} \end{aligned}$$

Other constructions (departing from the base  $\mathcal{AL}$  semantics) are the *union*:

$$(C \sqcup D)^I = C^I \cup D^I$$

*Full existential quantification*:

$$(\exists R.C)^I = \{a \in \Delta^I \mid \exists b.(a, b) \in R^I \wedge b \in C^I\}$$

(in this case arbitrary concepts are allowed to occur in the scope of the existential quantifier). *Number restrictions* expressing cardinality in the relations.

$$(\geq nR)^I = \{a \in \Delta^I \mid |\{b \mid (a, b) \in R^I\}| \geq n\}$$

and

$$(\leq nR)^I = \{a \in \Delta^I \mid |\{b \mid (a, b) \in R^I\}| \leq n\}$$



The *negation* of arbitrary concepts (going further than the negation of atomic concepts) is interpreted as:

$$(-C)^I = \Delta^I \setminus A^I C^I$$

All of this is no more than a fragment of first order predicate logic, and for the regards of this Thesis it is enough to know that there is a direct mapping from Description Logic expressions to first order predicate logic expressions; a more detailed account can be found in [25].

When representing the knowledge in one domain, a set of expressions will be given. These expressions may fall under the category of terminological axioms, which make statements about how concepts or roles are related to each other, and assertive axioms, which make statements about individuals, their attributes and their relations.

### T-Box and A-Box

The set of terminological axioms is often called T-Box. The most simple terminological axioms is the *inclusion* axiom:

$$C \sqsubseteq D \text{ (that is to say, } C^I \sqsubseteq D^I \text{)}$$

applied to two concepts  $C$  and  $D$ , and saying that concept  $C$  is a subset  $D$ , and the *equality* axiom,

$$C \equiv D \text{ (that is to say, } C^I \equiv D^I \text{)}$$

which says that the elements in  $C$  and  $D$  are the same. These constructors can also be used with roles.

When the left term in an equality is a concept, then it is said to be a *definition*. If each concept has at most one definition and there are no cycles in the definitions, then the resolution of the subsumptions is granted to be possible.

The world description or A-Box, is the set of assertions over individuals. Concepts and roles of the T-Box can be used, and defined individuals can be given attributes and relations to other individuals. If  $C$  is a concept and  $R$  is a role,

$$C(a) \text{ and } R(b, c)$$

says that the individual  $a$  belongs to the concept  $C$  and the individuals  $b$  and  $c$  are related through the role  $R$ . The first is a concept assertion, the second is a role assertion.

As it has been done with concepts and roles, an interpretation also maps individuals to the components of a domain,

$$a^I \in \Delta^I$$

## Reasoning

The following deductions (*inferences*) can be made from a model represented in Description logic:

- Class membership. To know whether an object is instance of a class or not.
- Classification. To know all the classes that an object belongs to.
- Equivalence of classes. To know if two classes have always the same individuals.
- Consistency of a class. To know if a class can have one or more individuals or will be always the empty set.
- Consistency of the ontology. If the ontology can have at least a set of instances which incur in no contradiction with the statements.

Note that there are other reasoning tasks which might be of interest:

- Explanation: reasoning task of providing axiom sets to explain a conclusion (what is important for editing and debugging).
- Conjunctive querying: check entailment of complex query patterns.
- Modularisation: extract sub-ontologies that suffice for (dis)proving a certain conclusion.
- Repair: determine ways to repair inconsistencies (related to explanation).
- Least Common Subsumer: assuming that class unions are not available, find the smallest class expression that subsumes two given classes.
- Abduction: given an observed conclusion, derive possible input facts that would lead to this conclusion.

## 3.3 The Semantic Web

### 3.3.1 The Semantic Web vision

Sir Tim Berners-Lee, acknowledged inventor of the World Wide Web and current director of the W3C (World Wide Web Consortium), envisioned the

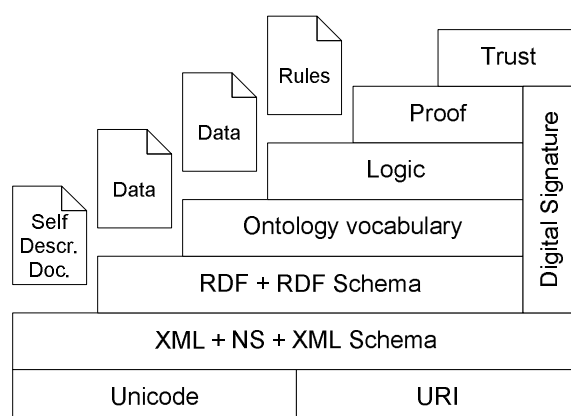


Figure 3.2: The Tim Berners-Lee's Semantic Web stack

*Semantic Web* in 1998 (an enjoyable and comprehensive exposition is given in [26]).

The vision of the Semantic Web was to set a universal medium for the exchange of data, in a broad sense. It envisaged smoothly interconnecting personal information management, enterprise application integration, and a global sharing of commercial, scientific and cultural data. Stress was made in that data should be in a machine-understandable form so that programs in the Internet could reach a certain degree of intelligence. It was said that the Web could reach its full potential only if it became a place where data could be shared and processed by both automated tools and by people.

Ten years after, the vision has been largely incarnated, although not entirely through the ‘machinization’ of data. A modern overview of the Semantic Web can be found in [27].

The Semantic Web vision defines a paradigm of technologies ordered in a hierarchical structure. The famous figure provided by Tim Berners-Lee in 2000<sup>4</sup> is shown in Figure 3.2.

Through the years, it has evolved and the technologies implementing the layers can be seen in Figure 3.3. The main components, found in Figure 3.3, are the Resource Description Framework (RDF) Core Model, the RDF Schema language (RDFs), the Web Ontology language (OWL), the SPARQL Protocol and RDF Query Language (SPARQL) and the Simple Knowledge Organization System (SKOS) (the latter not being in the figure but still an

<sup>4</sup><http://www.w3.org/2000/Talks/1206-xml2k-tbl/>

User Interface and Applications			Cryptography
Trust			
Proof			
Unifying logic			
Querying: SPARQL	Ontologies: OWL	Rules: RIF SWRL	
	Taxonomies: RDFS		
Data Interchange: RDF			
Syntax: XML			
Identifiers: URI		Character set: UNICODE	

Figure 3.3: The current Semantic Web stack

important and recent standard to be added to the stack).

These components scale in abstraction and complexity providing at least in theory with trust to the final applications. Trust is based in proofs, given in an unifying logic. The grammar to express this logic is OWL (implementing a Description Logic as it has been seen), which is an enriched evolution of taxonomies in RDFs. RDF triples, the atomic propositions are the most widely way to exchange information between Semantic Web applications. Data can be retrieved from RDF data stores with SPARQL queries, and the execution of rules can change the RDF data. XML is the syntax with which all these languages are written (SPARQL, OWL, SWRL, RDF). XML is based in a unified character set (UNICODE) and makes extensive use of URIs to uniquely reference the entities.

### 3.3.2 The Semantic Web technologies

In the lowest layers of the Semantic Web technology stack are found the URIs and XML.

#### Identifying resources: URI

A Uniform Resource Identifier (URI) [28] consists of a string of characters used to identify or name a resource.

URIs can be merely a name, in which case they are called URNs, Uniform Resource Names (like `urn:isbn:0-486-27557-4`, the unique name for a book), or can be the address where to retrieve it, in which case they are called URLs, Uniform Resource Locators (like `file:///home/username/-book.pdf`, the precise location of the book).

RFC 3986 provides the generic syntax to be used in all URI schemes. URI starts always with the *schema name* (an alphanumeric string starting with a letter), followed by a semicolon and a *hierarchical part* (composed by an authority followed by a path). Additionally, there may be a *query* (preceded by the question mark symbol ‘?’) and a *fragment* with additional information.

Besides RFC 3986 there are other identifying systems with similar properties. XRI, eXtensible Resource Identifier [29], is the OASIS response to this need, as well as ARK (Archival Resource Key) [30] generally used in archives.

But the most important, by far, is the DOI: Digital Object Identifier. The DOI<sup>5</sup> is a standard identifier for any intellectual property entity in the Internet, with wide acceptance in many contexts.

For the particular cases of identifying the different kinds of contents, there are of course other competing identification systems standards, but as with DOI, they are not free nor their use is reachable by everybody (only designated institutions can issue identifiers or can delegate their issue). These are the cases of ISBN (International Standard Book Number) for books, ISSN (International Standard Serial Number) for periodical publications, ISRC (International Standard Recording Code) for sound recordings, ISWC (International Standard Musical Work Code) for musical works, ISAN (International Standard Audiovisual Number) for audiovisual works, the ISMN (International Standard Music Number) for printed music etc. Other content identification formats have been described, like the TV-Anytime Content Reference Identifier (CRID) RFC 4078 but they are of marginal interest for this Thesis.

URI is the chosen representation of identifiers in all the systems developed for this Thesis, given its ease in use, its wide spread and the existing integration with the other tools employed. The MPEG-21 Digital Item Identification [31] does not define by itself any identification system but takes others and it is perfectly compatible with the URI.

---

<sup>5</sup>The International DOI Foundation: <http://www.doi.org/>

### Representing structured data: XML

XML (eXtensible Markup Language [32]) is the today's language for documents containing structured information. XML is used to encode documents and serialize data in a machine-readable format, being at the same time relatively human-legible. XML does not specify the semantics nor a set of tags, but only syntactic rules.

XML documents are composed of markup and content. There are six kinds of markup that can occur in an XML document: elements, entity references, comments, processing instructions, marked sections, and document type declarations.

Elements enclosed in angled brackets, and they can be start-tags (like `<contract>`), end-tags (like `</contract>`), and empty-element tags, (`<contract/>`). Content is simply the rest. Note that the word 'contract' is nowhere defined but by ourselves. The fact that the user has to define his own elements means that actually XML is a metalanguage rather than a language. There are hundreds of XML-based languages, but we can always define our owns. For example, in one imaginary language we might define, Listing 3.1 would be a valid document, representing the purchase of a CD by Alice from Bob.

Listing 3.1: A sample XML contract

```
<?xml version="1.0" encoding="UTF-8" ?>
<contract>
  <shopkeeper> Bob </shopkeeper>
  <customer> Alice </customer>
  <item "price=20"> SampleCD </item>
</contract>
```

Note that in Listing 3.1, the first line is an optional declaration of the language and that attributes can be given to elements (the price was an attribute of the element `item`<sup>6</sup>).

XML provides the possibility to define also structural relationships between the tags. All of the semantics of an XML document (excepting the hierarchy levels) will either be defined by the applications that process them or through the use of style sheets. It is therefore not true that XML is a syntax and not able to express semantics. It can, but not within a formal system, only to a limited extent, and subject to other rules defined by the applications.

---

<sup>6</sup>Note also that as a convention in this Thesis, fixed size font is used when referring to XML elements or other computer language expressions

There are different possibilities to express the semantic rules a document must follow like the XML Schema. According to this, there are two levels of correctness of an XML document:

- Well-formed. A well-formed document conforms to all of the XML's defined syntax rules.
- Valid. A valid document additionally conforms the semantic rules given in the XML Schema, DTD etc..

Section 4.1.3 analyzes more in depth the use of plain XML to represent contracts.

### A way of representing formats: XML Schema

The task of describing a language for representing contracts, would be as simple as declaring the grammar in one XML Schema file or in a Document Type Definition (DTD) or in Relax NG [33]. XML Schema is the preferred style in this Thesis; its official specification is given in [34] and [35].

XML Schemata are XML files themselves (a different representation could have been chosen, given that XML Schema is a metalanguage). The Listing 3.2 represents a simple grammar for contracts, where a trivial schema is proposed, and which allows to determine that the Listing 3.1 was a valid contract.

Listing 3.2: A sample XML Schema for contracts

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xs:element name="contract">
    <xs:annotation>
      <xs:documentation>a simple contract</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="shopkeeper"/>
        <xs:element name="customer"/>
        <xs:element name="item">
          <xs:complexType>
            <xs:complexContent>
              <xs:extension base="xs:anyType">
                <xs:attribute name="price" type="xs:decimal"/>
              </xs:extension>
            </xs:complexContent>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

        </xs:complexType>
    </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

It specifies the elements that are allowed in a contract document (like `customer`), in which order (for example `customer` cannot appear before `shopkeeper`), how many times they can appear, which attributes are valid (`price`) and which values they can take (`xs:decimal`).

Note that this is already an arbitrary task of modelling, and that many models of a representation are valid. For example, in Listing 3.2 there can only be one single item per contract. This could be improved by adding the attribute `maxOccurs = "unbounded"` to the `item` element declaration.

A visual representation of the example given by a popular XML editor is shown in Fig. 3.4. The information is provided symbolically: the three dots represent a sequence, the  $1..∞$  represents the cardinality of `item` (once corrected) etc. Some figures will be used following this convention; see the product's reference for more information<sup>7</sup>.

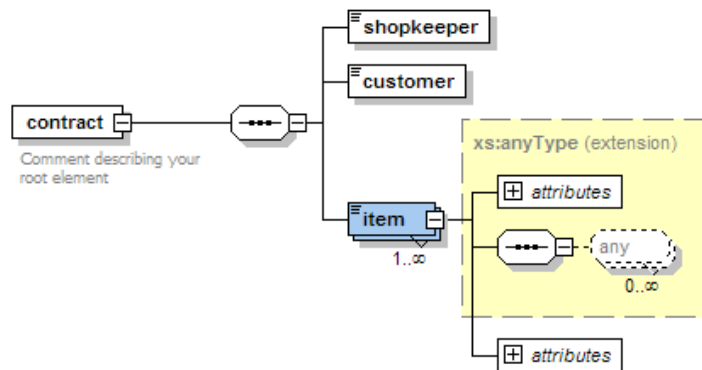


Figure 3.4: Visual representation of the contract XML Schema

### Identifying elements: namespaces

Very often a XML document contains elements defined in different schemata. After all, better than defining a new schema each time, is reusing existing ones. This way communication is easier and languages must not be defined

<sup>7</sup>Altova XML Spy. <http://www.altova.com>



<i>Pref.</i>	<i>Namespace</i>
Defined in this Thesis	
mvco	Media Value Chain Ontology <a href="http://purl.oclc.org/NET/mvco.owl#">http://purl.oclc.org/NET/mvco.owl#</a>
avc	Audiovisual Contract <a href="http://purl.oclc.org/NET/avc.xsd#">http://purl.oclc.org/NET/avc.xsd#</a>
From W3C Recommendations	
xsd	XML Schema <a href="http://www.w3.org/2001/XMLSchema#">http://www.w3.org/2001/XMLSchema#</a>
rdf	RDF <a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#">http://www.w3.org/1999/02/22-rdf-syntax-ns#</a>
rdfs	RDF Schema <a href="http://www.w3.org/2000/01/rdf-schema#">http://www.w3.org/2000/01/rdf-schema#</a>
owl	OWL <a href="http://www.w3.org/2002/07/owl#">http://www.w3.org/2002/07/owl#</a>
dc	Dublin Core <a href="http://purl.org/dc/elements/1.1/">http://purl.org/dc/elements/1.1/</a>
From MPEG standards	
dii	Digital Item Identification <a href="urn:mpeg:mpeg21:2002:01-DII-NS#">urn:mpeg:mpeg21:2002:01-DII-NS#</a>
r	REL Core schema <a href="urn:mpeg:mpeg21:2003:01-REL-R-NS#">urn:mpeg:mpeg21:2003:01-REL-R-NS#</a>
mx	REL Multimedia extension <a href="urn:mpeg:mpeg21:2003:01-REL-MX-NS#">urn:mpeg:mpeg21:2003:01-REL-MX-NS#</a>
sx	REL Standard extension <a href="urn:mpeg:mpeg21:2003:01-REL-SX-NS#">urn:mpeg:mpeg21:2003:01-REL-SX-NS#</a>

Table 3.1: Namespaces used in this document

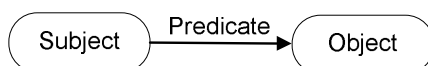


Figure 3.5: RDF triple

and learnt each time. For example, the element `xs:decimal` in Listing 3.2 is defined in the namespace of XML Schema whose reference is given in line 2 in the same listing.

To use XML Namespaces [36], a declaration is usually stated at the beginning of the document, and then a prefix followed by a colon is used. Table 3.1 shows the namespaces used throughout this document.

### Representing statements: RDF

The Resource Description Framework (RDF) [37] is a set of standards defining a method to represent simple *statements* with the structure of *subject*, *predicate*, *object* (Fig. 3.5). RDF is a key element of the W3C's Semantic Web activity, because information is very precisely and efficiently represented. The lack of further constraints has made it widely accepted in many applications. For the purposes of this Thesis, RDF is very appropriate for representing the propositions of the Predicate Logic described in Chapter 2. An extensive description of RDF is given in [38].

The only statements RDF can represent are called *triples*; and each of

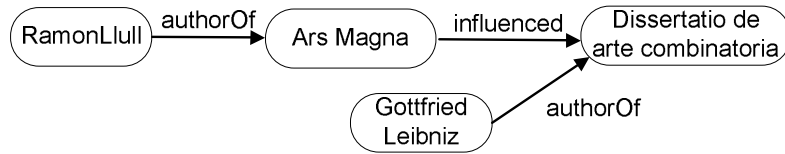


Figure 3.6: RDF example

their components (subjects, predicates or objects) are known as *resources*, uniquely identified by URIs. Objects can also be given as text values, called *literal values*, which may or may not be typed using XML Schema datatypes. Several RDF triples may refer to the same resource, and it is possible to create the logical graph this structure represents. Fig. 3.6 is the representation of three assertions.

Note that for XML or XML Schema, this Thesis has presented examples of its form as code because they were essentially a syntax. However, the importance of RDF is conceptual instead, and its actual materialization not unique. The concrete syntax in which RDF is codified as either Notation 3 (N3), Turtle [39] or XML [40]. Additionally, RDF can also be stored in files or RDF databases (triple stores).

### Representing structured resources: RDFs

RDFs (RDF Schema) [41], is an extensible knowledge representation language intended to structure RDF resources. It provides already the basic elements for the description of ontologies, with mechanisms for describing groups of related resources and the relationships between these resources. These resources are used to determine characteristics of other resources, such as the domains and ranges of properties.

The most elementary building block of RDFS is the *class*, which defines a group of individuals that belong together because they share some properties and which implements the class described in Section 3.2. The members of a class are known as instances, exemplars or individuals of the class. Classes are themselves resources and they are often identified by RDF URI references and may be described using RDF properties.

- `rdfs:Class` allows to declare a resource as a class for other resources.
- `rdfs:subClassOf` allows to declare hierarchies of classes.

In RDFs, instances are related to other instances through properties. A

RDF property is a relation between subject resources and object resources and is constrained by its range and domain.

- `rdfs:domain` of an `rdf:property` declares the class of the subject in a triple using this property as predicate.
- `rdfs:range` of an `rdf:property` declares the class or datatype of the object in a triple using this property as predicate.

Properties are also resources, (i.e., named by URIs) and may have sub-properties. Basic RDFs features are therefore:

- Classes and their instances;
- Binary properties between objects;
- Organization of classes and properties in hierarchies;
- Types for properties: domain and range restrictions.

One benefit of the RDF property-centric approach is that it allows anyone to extend the description of existing resources, one of the architectural principles of the Web. RDFs allows creating rich taxonomies, and it has already a rich expressivity. The Listing 3.3 shows a declaration, written in Turtle notation, of four classes (Persons, Authors, Books and Works), the hierarchical relationship between two of them (Book as a subclass of Work, Author as a subclass of Person), and a RDF property (`hasAuthor`), this time with the additional information that range and domain of the property are specified (the kind of resources that can act as object and subject of the property).

Listing 3.3: A sample RDF Schema

```
@prefix :      <http://www.example.org/sample.rdfs\#> .
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns\#> .
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema\#>.
:Person      a rdfs:Class .
:Work        a rdfs:Class .
:Book        a rdfs:Class; rdfs:subClassOf :Work .
:Author      a rdfs:Class; rdfs:subClassOf :Person .
:hasAuthor   a rdf:Property
              rdfs:range :Author;
              rdfs:domain :Work .
:RamonLlull  a :Author .
:ArsMagna    a :Book;
              :hasAuthor :RamonLlull .
```

### Querying the data: RDQL and SPARQL

The information found in a RDF store can be accessed through queries. The format of these queries can be made independent of the actual storage system, and either over a RDF database or file, a standard query can be launched. If XQuery [42] was the query language for XML, SPARQL is the query language for RDF.

RDQL (RDF Data Query Language [37]), was the precedent of SPARQL, and it was a query language specifically designed for RDF in 1998. RDQL queries have an XML-alike syntax, and they resemble that of SQL. For example, the following query retrieves those resources which have the property `requiresAuthorisation`:

Listing 3.4: A sample RDF query

```
<rdfq:rdfquery>
  <rdfq:From eachResource="http://dmag.upc.edu/mvco"/>
    <rdfq:Select>
      <rdfq:Property name="requiresAuthorisation"/>
    </rdfq:Select>
  </rdfq:From>
</rdfq:rdfquery>
```

Although RDQL was widely implemented by RDF frameworks, SPARQL has superseded it. SPARQL (SPARQL Protocol and RDF Query Language [43]) is a newer RDF query language, accepted as W3C Recommendation in 2008. It consists of triple patterns, conjunctions, disjunctions, and optional patterns. SPARQL can be used to express queries across diverse data sources, whether the data is stored natively as RDF or viewed as RDF via middleware. SPARQL contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions. The results of a SPARQL query can be results sets or RDF graphs.

An example of SPARQL sentence is given here. If an ontology defines the sentence *MakeAdaptation ResultsIn some Adaptation*, the query to ask which is the result of exercising a *MakeAdaptation*, will be:

Listing 3.5: A sample SPARQL query

```
PREFIX mvco: <http://dmag.upc.edu/mvco.owl#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?y
WHERE
{
  rrd:MakeAdaptation rdfs:subClassOf ?x .
  ?x owl:someValuesFrom ?y .
```

```
?x owl:onProperty rrd:ResultsIn .  
}
```

Note the use of three namespaces (given with PREFIX), the use of SELECT and WHERE similar to the SQL notation, and note that for this simple question, three different queries had to be crossed. *A priori* knowledge on the data model is needed.

It would be conceivable setting a server up, one which could answer to SPARQL queries. This could be done by using Joseki<sup>8</sup>. Joseki is an HTTP engine, developed within the W3C consortium, that supports the SPARQL Protocol and the SPARQL RDF Query language.

SPARQL Update [44] is the new proposal of W3C, still under study, to allow queries to perform writing besides reading operations (insertions, deletes, updates). SPARQL Update has not been used in this Thesis.

### Writing ontologies: OWL

We have just seen that RDFS provides, classes, class hierarchies, properties, property hierarchies and domain and range restrictions for them. But it does not provide property characteristics (inverse, transitive, etc.) nor complex concept definitions (in the sense of Section 3.2), nor cardinality restrictions or disjointness axioms.

Ontology Web Language (OWL) [45] became W3C Recommendation in 2004 [46], and it came to extend RDF Schema with the elements listed below.

- Logical expressions (and, or, not)
- (in)Equality
- Local properties
- Required/optional properties
- Required values
- Enumerated classes
- Symmetry, inverse attributes of object properties etc.

Between the objectives of OWL it was included having a layered language, with complex datatypes, digital signatures, local unique names, however, it was disregarded having arithmetic, procedures or default values for

---

<sup>8</sup>Joseki - A SPARQL Server for Jena. <http://www.joseki.org/>

Table 3.2: Expressivity of OWL sublanguages in version 1.0

<i>OWL Lite</i>	<i>OWL DL</i>	<i>OWL Full</i>
(sub)classes, individuals (sub)properties, domain, range conjunction (in)equality cardinality 0/1 datatypes inverse, transitive, symmetric properties someValuesFrom allValuesFrom	negation disjunction full cardinality enumerated types hasValue	metaclasses modify lang.

the properties. What was desired was having one version implementing a decidable Description Logics: OWL-DL (and a reduced version called OWL-Lite). The three levels in the language are therefore, in increasing order of complexity *OWL Lite*, *OWL DL* and *OWL Full*.

OWL Lite adds over RDFS simple constraints and hardly a few other constructs. It lacks explicit negation or union, cardinality is restricted to 0/1 and no nominals can be used (`oneOf`). OWL Lite is equivalent to a  $SHIN(\mathcal{D})$  Description Logics<sup>9</sup>. OWL Lite can be considered a failure: it is not really expressive but it is almost as complex as OWL DL (problems can be as problem as to require an exponential time), given that a complex syntax hides real expressive power. Its current use in ontologies is accidental rather than intentional.

OWL DL is equivalent to a  $SHOIN(\mathcal{D})$  Description Logics<sup>10</sup> and it gives the maximal expressiveness while keeping tractability.

OWL Full has even greater expressiveness but it loses tractability and eventually reasoning cannot be performed at all. The features of each level can be seen in Table 3.2.

An OWL ontology in the abstract syntax contains a sequence of *axioms*, *facts*, and *annotations*. This naming might be confusing, as these words are not used in the sense widely accepted in logic sciences.

Annotations on OWL ontologies are merely metadata, or comments, and they can be used to record authorship and other information associated with an ontology, including imports references to other ontologies. The OWL

<sup>9</sup>The notation  $SHIN(\mathcal{D})$  describes the capabilities of the language, where  $\mathcal{S}$  is equivalent to  $\mathcal{ALC}$ , attributive language with complex concept negation,  $\mathcal{H}$  stands for the fact of having role hierarchies,  $\mathcal{I}$  for inverse properties,  $\mathcal{N}$  for cardinality restrictions, and  $(\mathcal{D})$  for datatype properties

<sup>10</sup>Additionally,  $\mathcal{O}$  stands for nominals, enumerated classes of object value restrictions

Table 3.3: OWL constructs

OWL constructs	OWL Construct	DL Example
intersectionOf	$C_1 \sqcap \dots \sqcap C_n$	<i>Thesis</i> $\sqcap$ <i>English</i>
unionOf	$C_1 \sqcup \dots \sqcup C_n$	<i>English</i> $\sqcup$ <i>Catalan</i>
complementOf	$\neg C$	$\neg$ <i>English</i>
oneOf	$o_1, \dots, o_n$	<i>June, September</i>
allValuesFrom	$\forall P.C$	$\forall$ <i>hasLanguage.English</i>
someValuesFrom	$\exists P.C$	<i>exists</i> <i>hasLanguage.English</i>
value	$\exists P.o$	$\exists$ <i>hasDate. June</i>
minCardinality	$\geq nP.C$	$\geq 5$ <i>hasPart.Chapter</i>
maxCardinality	$\leq nP.C$	$\leq 100$ <i>hasPage.Chapter</i>
cardinality	$= nP.C$	$= 5$ <i>hasMember.Committee</i>

Table 3.4: OWL axioms

OWL	Axiom DL	Example
subClassOf	$C_1 \sqsubseteq C_2$	<i>Human</i> $\sqsubseteq$ <i>Animal</i> $\sqcap$ <i>Biped</i>
equivalentClasses	$C_1 \equiv \dots \equiv C_n$	<i>Man</i> $\equiv$ <i>Human</i> $\sqcap$ <i>Male</i>
subPropertyOf	$P_1 \sqsubseteq P_2$	<i>hasDaughter</i> $\sqsubseteq$ <i>hasChild</i>
equivalentProperties	$P_1 \equiv \dots \equiv P_n$	<i>cost</i> $\equiv$ <i>price</i>
sameIndividual	$o_1 = \dots = o_n$	<i>PresidentBush</i> = <i>GWBush</i>
disjointClasses	$C_i \sqsubseteq \neg C_j$	<i>Male</i> $\sqsubseteq$ $\neg$ <i>Female</i>
differentIndividuals	$o_i \neq o_j$	<i>john</i> $\neq$ <i>peter</i>
inverseOf	$P_1 \equiv P_2^-$	<i>hasChild</i> $\equiv$ <i>hasParent</i> <sup>-</sup>
transitive	$P^+ \sqsubseteq P$	<i>ancestor</i> <sup>+</sup> $\sqsubseteq$ <i>ancestor</i>
symmetric	$P \equiv P^-$	<i>connectedTo</i> $\equiv$ <i>connectedTo</i> <sup>-</sup>

constructs to express facts are given in Table 3.3.

Facts state information about a particular individual, in the form of classes that the individual belongs to as well as properties and values of that individual. Facts are also used to establish whether two individuals are the same or different.

Axioms (formerly called definitions) are used to associate class and property identifiers with either partial or complete specifications of their characteristics (like saying if a property is symmetric), and to give other information about classes and properties (for example, if a class is a subclass of another). The axioms shown in Table 3.4 can also be expressed.

OWL ontologies have a unique name, and it is usually an URI pointing to the actual location of the ontology, although this is not part of the formal meaning of OWL. Imports annotations are directives to retrieve a Web document and treat it as an OWL ontology.

For example, to show its suitability for the objectives of this Thesis, the OWL expression (given in RDF/XML syntax) in Listing 3.6 states that to create a work no other previous IP entity is needed. For doing this, the class `CreateWork` is subclassed from an expression:

$$\text{CreateWork} \subseteq \neg ( \exists (\text{actedOver IPEntity}) )$$

Listing 3.6: OWL fragment in RDF/XML syntax

```
<owl:Class rdf:about="\#CreateWork">
  <rdfs:subClassOf>
    <owl:Class>
      <owl:complementOf>
        <owl:Restriction>
          <owl:onProperty>
            <owl:ObjectProperty rdf:about="\#actedOver"/>
          </owl:onProperty>
          <owl:someValuesFrom rdf:resource="\#IPEntity"/>
        </owl:Restriction>
      </owl:complementOf>
    </owl:Class>
  </rdfs:subClassOf>
```

Note that OWL does not explicitly separate the ABox from the TBox. Thus, it is possible to find in the same OWL fragment expressions proper of the ABox, like the statement of Listing 3.7 which simply declares that Alice is a user.

Listing 3.7: Individual declaration in OWL

```
<rdf:Description rdf:about="http://purl.oclc.org/NET/
mvco.owl\#Alice">
  <rdf:type rdf:resource="http://purl.oclc.org/NET/
mvco.owl\#User"/>
```

## The new ontologies: OWL 2

OWL 2 became a formal W3C Recommendation on 27 October, 2009 [47]. OWL 2 adds new functionality with respect to OWL 1. Some of the new features are syntactic shortcuts to already expressible terms while others offer new expressivity, including keys, property chains, richer datatypes, data ranges, qualified cardinality restrictions, asymmetric, reflexive, and disjoint properties and enhanced annotation capabilities.

OWL 2 introduces also the concept of *OWL profiles*. OWL 2 Profiles are sub-languages (syntactic subsets) of OWL 2 that offer important advantages in particular scenarios: OWL 2 EL, OWL 2 QL, and OWL 2 RL. The three



of them are subsets of OWL DL which can be solved in polynomial time, maximizing one or other aspect in the expressivity.

OWL 2 EL enables polynomial time algorithms for all the standard reasoning tasks; this is particularly interesting for applications where very large ontologies are needed. OWL 2 QL enables conjunctive queries to be answered using standard relational database technology, interesting to launch SQL queries. OWL 2 RL enables the implementation of polynomial time reasoning algorithms using rule-extended database technologies operating directly on RDF triples; it is particularly suitable for applications where relatively lightweight ontologies are used to organize large numbers of individuals and where it is useful or necessary to operate directly on data in the form of RDF triples.

### Writing rules: SWRL

The abstract concept of rules may refer to *logical rules* (the implication seen in predicate logic), to *procedural rules* (an event triggers a set of instructions that are executed). Rules as understood here are of the first kind, simply first-order logic implications. A more precise definition of logical rule is that of a Horn clause: a Horn clause is every statement with at most one positive literal (only one atomic expression not negated).

The following is an example of a (definite) Horn clause (where  $u$  is the only positive literal):

$$\neg p \vee \neg q \vee \dots \vee \neg t \vee u$$

which can be simply rewritten equivalently in the form of an implication:

$$(p \wedge q \wedge \dots \wedge t) \rightarrow u$$

The terms of the implication are called *antecedent* (or body) and *consequent* (or head), where the consequent is inferred from the antecedent. Therefore, rules present the following general form:

$$\textit{antecedent} \rightarrow \textit{consequent}$$

Semantic Web Rule Language (SWRL) [48] is the language of the Semantic Web to represent rules, although it is not yet a full standard and it is only a candidate. It proposes adding rules as new constructs of the OWL language (under the form of axioms).

In SWRL, both antecedent and consequent are said to be made of *atoms*. Atoms can be  $C(x)$ ,  $P(x, y)$ ,  $\textit{sameAs}(x, y)$ ,  $\textit{differentFrom}(x, y)$ , or  $\textit{builtIn}(r, x, \dots)$  where  $C$  is an OWL description or data range,  $P$  is an OWL property,  $r$  is a built-in relation,  $x$  and  $y$  are either variables, OWL individuals or OWL data values, as appropriate.

Variables are indicated using the standard convention of prefixing them with a question mark (i.e.  $?x$ ). The example of a rule is:

$$isAuthor(?x1, ?x2) \wedge hasLanguage(?x2, ?x3) \rightarrow canSpeak(?x1, ?x3)$$

Listing 3.8 shows the codification of the previous example in SWRL. This code could be inserted within a larger OWL file declaring the classes *Book*, *Author* and *Language* and the object properties *hasAuthor*, *hasLanguage* and *canSpeak*. Note the verbosity of 3.8 compared with the expression above: it lacks the elegance of Prolog expressions but integrates well with the Semantic Web framework.

Listing 3.8: Sample SWRL rule

```
<swrl:Imp rdf:ID="AuthorsLanguage">
  <swrl:head>
    <swrl:AtomList>
      <rdf:first>
        <swrl:IndividualPropertyAtom>
          <swrl:propertyPredicate rdf:resource="#canSpeak"/>
          <swrl:argument2>
            <swrl:Variable rdf:ID="x3"/>
          </swrl:argument2>
          <swrl:argument1>
            <swrl:Variable rdf:ID="x1"/>
          </swrl:argument1>
        </swrl:IndividualPropertyAtom>
      </rdf:first>
    </swrl:AtomList>
  </swrl:head>
  <swrl:body>
    <swrl:AtomList>
      <rdf:rest>
        <swrl:AtomList>
          <rdf:first>
            <swrl:IndividualPropertyAtom>
              <swrl:propertyPredicate rdf:resource="#hasLang"/>
              <swrl:argument2 rdf:resource="#x3"/>
              <swrl:argument1>
                <swrl:Variable rdf:ID="x2"/>
              </swrl:argument1>
            </swrl:IndividualPropertyAtom>
          </rdf:first>
        </swrl:AtomList>
      </rdf:rest>
      <rdf:first>
        <swrl:IndividualPropertyAtom>
          <swrl:argument2 rdf:resource="#x2"/>
          <swrl:propertyPredicate rdf:resource="#isAuthor"/>
          <swrl:argument1 rdf:resource="#x1"/>
        </swrl:IndividualPropertyAtom>
      </rdf:first>
    </swrl:AtomList>
  </swrl:body>
</swrl:Imp>
```

```

        </swrl:IndividualPropertyAtom>
    </rdf:first>
</swrl:AtomList>
</swrl:body>
</swrl:Imp>

```

Even though SWRL is not a W3C recommendation, there are plenty of applications making use of it and implementing partially the proposal. In practice, reasoners do not support the full specification because in the way SWRL is specified, the reasoning becomes undecidable.

Different implementations have followed different approaches to build a reasoner. For example, Hoolet<sup>11</sup> translates SWRL into first order logic and then uses a theorem prover, while Pellet<sup>12</sup> expands the tableaux algorithms of the OWL-DL reasoner.

### Mapping concepts: SKOS

Simple Knowledge Organization System (SKOS) [49] although a new standard (W3C Recommendation only from August 2009) is not a step beyond OWL, but rather simpler in ambitions and complexity.

SKOS is a common data model for sharing and linking knowledge organization systems via the Web. SKOS provides elements to share terminologies, classification schemes and in general knowledge organization systems from different origins. It also provides a lightweight language for developing and sharing new knowledge organization systems. SKOS may be used on its own or in combination with OWL.

SKOS sees the *concept* as the unit of thoughts. From each knowledge organization system, existing concepts can be asserted simply in a line:

```
mvco:Thesis rdf:type skos:Concept
```

New concepts are expected to be given as HTTP URIs. The same concepts may have different labels, with one preferred and several alternatives (e.g. it can be asserted that `PhDDissertation` denotes Thesis too).

The key idea in SKOS is that concepts can be semantically related. This semantic relationship is not of the rigorous kind of OWL, where a logical formalism was behind to forbid wrong expressions. SKOS does not pretend to embody any formal language and thus, although semantic relations can be established, inconsistencies are tolerated in the assertions and no consistency validators are expected to be run.

<sup>11</sup>Hoolet reasoner <http://owl.man.ac.uk/hoolet/>

<sup>12</sup>Pellet OWL DL reasoner <http://pellet.owldl.com>

The main relationships are of the kind broader/narrower. A concept is the same, or narrower or broader than other (or not related at all). However SKOS does not cover the distinction between types of hierarchical relation: for example, instance-class and part-whole relationships. In addition, concepts can be associated to other concepts.

Whole vocabularies are represented as *concept schemes*, SKOS provides the elements to establish mappings between these schemes by giving elements like `exactMatch`, `broadMatch` or `relatedMatch`.

### 3.4 Practical use of Ontologies

#### Ontology access libraries

Representation and access to the representation is well separated in ontology technologies. Section 3.3 has described the semantics of Description Logic in the Semantic Web, as well as its syntax. OWL ontologies can be serialized as XML -a completely neutral technology-. However, it is not practical editing text files with the adequate sentences. In real life, there are other ways for doing this, either full applications or libraries callable from computer programs.

The edition of ontologies during the research work for this Thesis has been carried out with the Protégé editor<sup>13</sup>, although many other editors exist (like the TopBraid Composer<sup>14</sup>, SWOOP<sup>15</sup>, the Altova Semantic Works<sup>16</sup>, ModelFutures OWL Editor<sup>17</sup> or the NeOn Toolkit<sup>18</sup>). Protégé is the most widely spread editor and serves also to visualize the ontology, but other specific applications to visualize ontologies do exist (for a general survey see [50]).

Of more interest for research and development purposes are the libraries and APIs existing to be called from the applications. Currently there are three major libraries which offer access to the OWL data, but they are actually very similar. The three are Java based and offer approximately the same functionality: the *Protégé library*, the *OWLAPI* project, and the *Jena API*.

---

<sup>13</sup>The Protégé Ontology Editor and Knowledge Acquisition System, <http://protege.stanford.edu>

<sup>14</sup>TopBraid composer, <http://www.topbraidcomposer.com/>

<sup>15</sup>SWOOP, Semantic Web Ontology Editor, <http://code.google.com/p/swoop/>

<sup>16</sup>Semantic Works Visual RDF and OWL editor, <http://www.altova.com/semanticworks.html>

<sup>17</sup>The Model Futures OWL Editor, <http://www.modelfutures.com/owl>

<sup>18</sup>NeOn Toolkit, <http://neon-toolkit.org/>

Jena<sup>19</sup> is the most widely used Java library. It was started by Hewlett Packard but it has currently an Open License. Jena is the API chosen for this Thesis, justified by the maturity of the software, the spread of its use and its ease of use. It is still an active project and the last version (2.5) dates from December 2008.

Jena is a broader framework providing a RDF API, and OWL API, capabilities to read and write RDF in RDF/XML or in N3, to keep complex memory models of information and to process SPARQL queries and an own-defined kind of rules.

The Jena OWL API gives access to OWL providing a set of helper Java classes and functions to manipulate the ontology in a very similar to other OWL access libraries. Jena is designed as an overly complex system, poorly documented and whose efficient access requires knowing the ontology structure. Anyway Jena has useful classes to represent graphs, resources, properties and literals, and at least they facilitate the tasks of accessing the ontology.

### Ontology reasoning

Reasoners are the applications able to perform any of the reasoning described in Section 3.2.1. They can be directly invoked as regular applications, they can be invoked as libraries from an API and they can be invoked by calling one of the standard services they usually offer through the HTTP protocol. This standard protocol is called the 'DIG interface' (DIG is short for DL Implementation Group). A DIG compliant reasoner is a Description Logic reasoner that provides a standard access interface (a.k.a. the DIG interface), which enables the reasoner to be accessed over HTTP, using the DIG language.

API access to the reasoner is formally decoupled from the APIs to access the OWL, although there are some combinations of libraries specially supported: the reasoner Pellet<sup>20</sup> integrates well with Jena code and for this reason has been chosen as the working reasoning library for our project. Pellet is open source and it has a MIT license, but still it is in the cutting edge of technology: its current version supports already OWL 2 and the most recent features like fuzzy reasoning and probabilistic reasoning.

Other reasoners disputing Pellet supremacy are RacerPro<sup>21</sup>, FACT++<sup>22</sup>

---

<sup>19</sup> Jena <http://jena.sourceforge.net>

<sup>20</sup> Mindswap Pellet, <http://pellet.owldl.com/>

<sup>21</sup> RacerPro commercial reasoner <http://www.racer-systems.com/>

<sup>22</sup> FaCT++ GNU Reasoner <http://owl.man.ac.uk/factplusplus/>

<i>OWL Editors</i>	<i>APIs</i>	<i>Reasoners</i>
Protégè	Jena	Pellet
TopBraid Composer	Protégè API	Hoolet
SemanticWorks	OWLAPI	Fact++
NeOn Toolkit		Racer Pro
SWOOP		HermiT

Table 3.5: Some semantic web tools

or more recently HermiT<sup>23</sup>.

One of the major critics posed to Description Logics was the lack of efficient implementation of their reasoners. This problem has been overcome with a notable improvement in the reasoning algorithms (like the hyper-tableaux in HermiT) and with the structural changes in OWL 2. Additionally extremely large datasets (like the metadata in the Europeana project<sup>24</sup>) can also be ported to Prolog triples and reason with them [51]. All of this, makes the efficiency no longer a valid excuse not to use ontologized data. Some reasoners and other Semantic Web tools are enumerated in Table 3.5, the first row showing the tools chosen for this Thesis.

<sup>23</sup>HermiT reasoner, <http://www.hermit-reasoner.com>

<sup>24</sup>Online collection of digitized items from European museums, libraries, archives and multi-media collections. <http://www.europeana.eu>

## Chapter 4

# Contract representation

This Chapter describes the state of the art on multimedia agreements representation. For this, a review of the evolution of contract representation is given, paying attention to both from industrial and academic initiatives. Then, the value chain representations, central for this Thesis, are reviewed. For illustration purposes a sample audiovisual contract is provided with a succinct description.

This work is neither the first endeavour towards a computerized representation, management and execution of contracts (more than fifty years of work precede) nor the first system specifically handling automatically multimedia content (Digital Rights Management systems have also been running for at least fifteen years). Contract representation will be analyzed in this Chapter (Section 4.1) with some detail and Digital Rights Management systems in the next Chapter. Value Chain representations, necessary to describe the different types of multimedia contracts are also described in this Chapter (Section 4.2)

### 4.1 Digital Representation of Contracts

#### 4.1.1 Objectives of a contract representation

An agreement is a mutual promise between two or more parties. A contract is defined as ‘a promise enforceable by law’ by the Encyclopedia Britannica, but very often it refers to the representation of the agreement too. Contracts are made of mutual promises between two parties to do (or refrain from doing) something. The terms of a contract may be expressed written or

orally, implied by conduct, industry custom, and law or by a combination of these.

Agreements have had a fixed representation since the invention of script several millennia ago and written agreements have been found from the earliest steps of the writing development. The famous Hammurabi code, carved in diorite stone at some time thirty centuries ago, is the first remarkable public law document preserved, but personal agreements, belonging to the private law documents sphere, appeared roughly at the same time. To highlight the relevance of this achievement, it can be noted that ordered trade fostered economical growth, and civilizations that adopted written documents developed much faster: the same principle probably still holds, and more ordered and fair business at the end benefits the society as a whole. Digital expression of agreements is a step further in this quest that started thirty centuries ago.

Having a permanent representation of the agreement had in that time a clear purpose: to have an undisputed authority reminding the contract terms in case of conflict. For a very long time nothing seemed to have changed and contracts from the very early times to the second half of the XXth century had remained similar in purpose (to keep a record) and simplicity.

The major disruption in this contract tradition started when the contract scope started broadening and becoming more complex. At the beginning, contracts usually focused on a single event, like a marriage, an inheritance or the purchase of a land, and the paradigmatic contract was a snapshot taken at the moment a bargain was made.

The twentieth century witnessed the development of a modern contract law that largely overthrew classical contract law, and many contracts became more complex in nature and began being tracked actively. Some of them were not about an event but about a process. For example, it is not uncommon in multimedia agreements paying royalties whose exact amount depends on a number of factors, and whose execution happen periodically (e.g. royalties payment is delivered monthly). Also, contracts started referring to multiple events, and started taking into consideration several scenarios expanding the temporal range of the contract. So, for example, static rules of interpretation were replaced by dynamic rules that took into account events before and after the moment of contract formation. In this context, and with the appearance of computers, contracts started becoming *active* when they were represented as computer formulas, spreadsheets or other active documents. A complete and structured description of the improvements and the new business opportunities introduced by e-contracting are presented in [52].



<p>Video excerpt License Agreement  LICENSOR (referred to as Distributor) - on one part - and LICENSEE  (referred to as User) jointly referred to as the Party/Parties)</p> <p>WHEREAS:</p> <p>(a) User has requested permission of Distributor to use news excerpts about blackout in Italy taken from the archives of RAI - Radiotelevisione Italiana S.p.A. ("Owner"), as specified hereunder ("Archive");</p> <p>(b) Distributor has been authorized by Owner to handle distribution of Owner's</p> <p>(c) User intends to acquire, with respect to Archive or part thereof, non-exclusive Archive; rights ("Rights") for the territory of ("Territory"), subject to the terms and conditions set forth in this Agreement.</p> <p>Pursuant to points (a), (b) and (c) above, User and Distributor hereby agree to the following:</p>
--

Table 4.1: Sample contract (I preamble)

In the previous example, the formula calculating the royalties' exact amount can be in a spreadsheet document, and this document can be the basis for negotiation and signature. (It has to be noted that not every computer representation of a contract is active. For example, an old diskette storing a narrative contract written in Word Perfect is still a passive contract.

From the point view of this Thesis (and with some nuances also of [53]), the major revolution on the contract representation is that passive documents aimed at reminding what had been agreed became steering elements in the parties' relationship. In the case of active e-contracts, a framework must be defined for the operation based on it. A general framework can be found in [54]; for some particular languages a framework description follows.

#### 4.1.2 A sample audiovisual contract

Audiovisual contracts in this Thesis refer to contracts in the audiovisual market. For the purposes of illustration, a sample contract is shown in Table 4.1, where a verbose copy has been made -excepting the data affecting the privacy of the parties.

Preamble of a contract contains some metadata information on the contract, like a date, the title of the contract, a definition of the parties, or even the object of the contract.

<p>1. Distributor shall deliver to User, with respect to Archive, one or more Betacam SP videocassettes in the PAL broadcast standard ("Material") containing Archive on an "as is" basis.</p> <p>2. Distributor grants to User, with respect to Archive, the Rights in the Territory, both as specified in point (c) above. User shall exploit the Rights from the date of execution of this Agreement for ... (...) year ("License Period").</p> <p>3. User shall notify Distributor of any claim regarding the technical conditions of Material within and no later than 10 (ten) days from collection of Material by User telecopier transmission, the deadline for claims being essential to this Agreement. If Material should not prove technically satisfactory, Distributor's action shall be limited to replacement of Material at no cost for the User.</p> <p>4. The Distributor and the Owner do not undertake any liability with regard to the technical conditions and/or the quality of the Archives and Material. The Archives and the relevant material are supplied as they are and without guarantees of any kind by the Distributor and the Owner [...]</p> <p>5. User shall use Archive exclusively in a program entitled XXX and limited to the Rights and Territory granted hereunder. Any other use of Archive is expressly prohibited with respect to this Agreement, but may be subject to further agreements between Distributor and User. Notwithstanding the foregoing, Distributor and Owner shall be free to use Archive worldwide during the License Period, and User expressly acknowledges and accepts that the Rights are granted to User on a non-exclusive basis.</p> <p>6. User expressly warrants that Archive shall not be used to libel, slander, or in any other way damage the moral integrity, dignity and/or image of the persons, places, companies and/or any other elements portrayed in Archive, and that User shall not use a music score with Archive which is inconsistent with, or may in any way cause damage to, the persons, places, companies and/or any other elements portrayed in Archive. Notwithstanding the foregoing, User holds Distributor harmless against any claim by third parties with respect to improper use of Archive.</p>
--

Table 4.2: Sample contract (II, clauses 1-6)

<p>7. User shall not exploit elements in the Archive, if any, which are owned by third parties, including but not limited to photographs, transparencies, excerpts from motion pictures, and/or theater plays, videograms or other elements expressly copyrighted said third parties. User holds Distributor harmless against any liability in by this respect.</p> <p>8. User shall pay local performing rights societies in the Territory for music performance rights relating to music, if any, included in, or added by the User to, Archive.</p> <p>9. Upon the earliest use of Archive, User shall notify Distributor of the actual amount of minutes of Archive used, within and no later than 20 (twenty) days from the delivery of the Material. Failing a positive reply on User part, Distributor will be forced to issue an invoice for 5 (five) minutes used as guarantee for the delivered Material. User shall also provide Distributor with a VHS recording of the program where Archive was used, and shall return any unused Material.</p> <p>10. In consideration of the copyrights materials granted in this Agreement and additional fees, User shall pay Distributor a license fee of Euro ... (...) gross, per minute or part thereof used. User states that, according to the ...-Italy Convention for the avoidance of double taxation, the with ... (... per cent) of the gross amount of the license fee. Therefore User shall pay Distributor a license fee of Euro ..., net. The amount above shall be paid by User upon receipt of invoice by check or bank transfer made payable to: LICENSOR ...</p> <p>11. User and Distributor hereby agree that the applicable law with respect to the interpretation and/or performance of this Agreement shall be Italian Law and elect the Forum of Rome as the exclusive Forum to hear any disputes pertaining to this Agreement.</p>
---

Table 4.3: Sample contract (III, clauses 7-11)

### 4.1.3 Overview of digital representations of contracts

Efforts to represent contracts electronically are not new -they are as old as computers, and even making them part of digital systems is not new. Along with the development of computer sciences and network communications, the electronic representation of contracts played each time a more active role. Thus, in the earliest Electronic Data Interchange (EDI) standards, about forty years ago, only bills and invoices were exchanged, but slowly the exchanged messages became richer in their expressivity and their role in an integrated information system was each time more important.

Besides proprietary systems where information acquired an ad-hoc structure there have been some remarkable attempts to structure electronically the information in contracts. COSMOS [55] was an e-commerce architecture supporting catalogue browsing, contract negotiation and contract execution. It defined a contract model in UML and proposed a CORBA-based software architecture in a coherent manner. DocLog [56] was an electronic contract representation language introduced in the 2000 with a 'XML like' structure, which anticipated the next generation of XML-based contract representations. When XML was mature enough it was seen as a good container of contract clauses, and thus the new format specifications came under the form of a XML Schema or a DTD (Document Type Definition). An effort to achieve a common XML contract representation was the Contract Expression Language (CEL) [57] in 2004, developed by the Content Reference Forum.

In the following years, the advent of the Semantic Web reached the contract expression formats, and new representations evolved from the syntactic representation level to the semantic one ([58] [59] [60]) being developed domain ontologies in the KIF (Knowledge Interchange Format) or OWL (Ontology Web Language) languages. Some of these contract models, sometimes using rules [61] have also been aimed at governing Information Technology systems [62] [63], but never achieved wide acceptance. Among them, it is worthy to mention SweetDeal [64], a rule-based approach to representation of contracts that enabled software agents to create, evaluate, negotiate, and execute contracts with substantial automation and modularity.

Another general oriented contract language with aspirations of formality was the Business Contract Language (BCL). *The aim of BCL is to provide sufficient expressive power to describe contracts, including conditions which specify real-time processing, yet be simple enough to retain a human oriented style for expressing contracts* [65]. BCL had to be understood amidst a more complex architecture model, Web Service oriented, to support the cross-

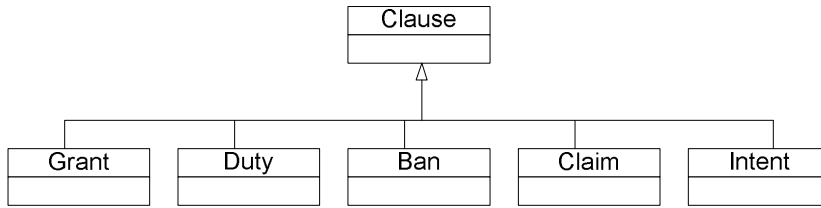


Figure 4.1: Classification of clauses in the CEL model

organizational nature of collaborations and to integrate contract management services into the overall business processes between organizations. The BCL expressed the semantics of contracts in this context, and as most notable refinements it has to be noted the initiative of [21] to provide RuleML expressions in the formalization. By using defeasible logic, clauses could be to some extent contradictory while keeping flexibility.

Nevertheless, XML representation of contracts is not a past theme but a topic of active research [66], and the newest and two most promising alternatives for representing contracts is again XML based: CEL and eContracts. For the interest for this Thesis, the next paragraphs describe more in detail these CEL and eContracts formats.

#### 4.1.4 Contracts Expression Language

CEL formalizes a language that enabled machine-readable representation of typical terms found in content distribution contracts and it is compliant with the Business Collaboration Framework [67].

For the purposes of this work, it should be highlighted the bright approach of CEL to the classification of clauses present in a contract based on the deontic logic. The deontic logic expresses *claims*, *duties*, *bans* and *possibilities*. Clauses can be classified systematically attending at if they represent a *is*, a *must*, a *must not* or a *may*, and if the verb applies to one party or to the other. Now it becomes manifest the interest of the review of deontic logic given in Section 3.1.3, nevertheless a more detailed analysis in deontic logic for contracts can be found in [68]. Taking this into account, CEL clauses are classified as shown in Fig. 4.1.

In CEL, a contract is conceptually a collection of promises agreed to by its signing parties. For them a promise is a collection of clauses issued by its signing parties and a clause describes a relationship among an *event*, a *principal*, an *act*, a *resource* and a *condition*. Fig. 4.2 shows the structure of a contract expressed in the CEL language, and Listing 4.1 a sample CEL

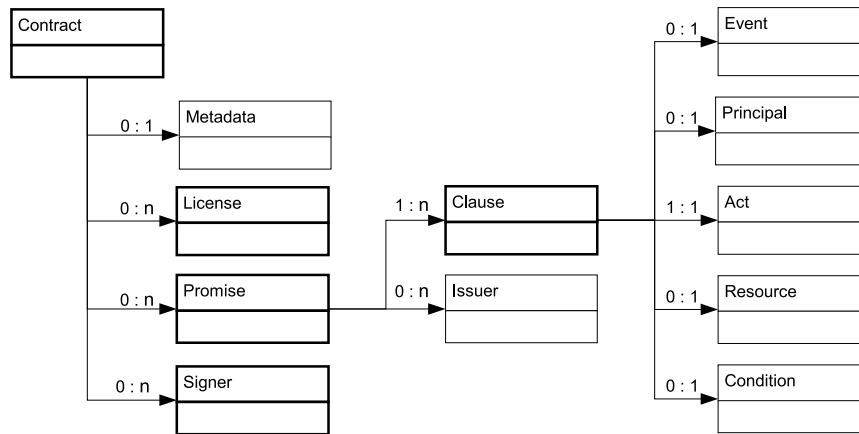


Figure 4.2: The CEL contract model.

contract.

It is very interesting the idea of modeling *events* in the contract, and the work presented here will further investigate it.

Listing 4.1: Sample contract in CEL format

```

<?xml version="1.0" encoding="utf-8"?>
<contract>
  <promise>
    <duty>
      <refx:receiveCR/> <r:keyHolder licensePartId="Bob">
        <refx:redirect> ... </refx:redirect>
      <refx:request licensePartId="CR"/>
      <refx:requestConstraint>
        <r:xmlExpression>/TYPE="phd" </r:xmlExpression>
      </refx:requestConstraint>
    </duty>
  </promise>
  <signer licensePartId="Bob"></signer>
  <signer licensePartId="Alice"></signer>
</contract>

```

Although CEL is an easy language (it departs minimally from MPEG21-REL), and it is well built, no news come from CEL what reveals that not many projects are considering it. The next paragraphs describe a new and more promising alternative.

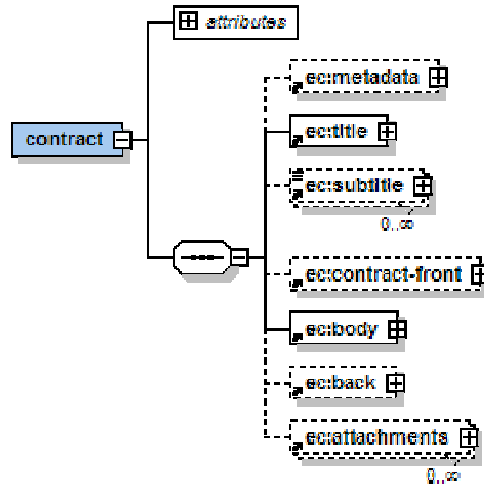


Figure 4.3: Top elements in the eContracts XML contract

#### 4.1.5 OASIS eContracts

Currently, the most widely acknowledged standard in the digital contracts area is the eContracts, promoted by the OASIS consortium. This electronic contract representation banks also on XML, but it has gained rapid acceptance. This is the culmination of the LegalXML eContracts Technical Committee, which started in 2002 to evaluate a possible eContracts Schema (see the evolution in [69]), and which achieved its final form in 2007 [70]. The model proposed in this chapter uses this eContract standard as a framework for the execution of contracts in the audiovisual B2B sector. Being eContracts the container, contents can be expressed with the help of the Media Value Chain Ontology (object of this work and to be described in Chapter 6) and the principles of deontic logic for contracts execution in other formalizations.

eContracts is a standard aimed at representing general contracts, having no particular field in scope. eContracts documents are composed of general paragraphs and clauses, being the main XML elements the `ec:item`, the `ec:title`, the `ec:block` and the `ec:text` with the item element used recursively.

Figure 4.3 shows the XML schema for the root and the main elements (the root element is, of course, the `ec:contract`). `ec:metadata` elements in the contracts allow the specification of contract date, creator or title (using the Dublin Core metadata elements). Contract parties are declared in the

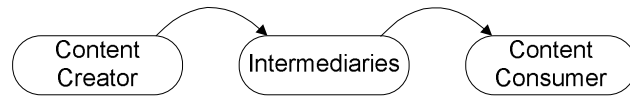


Figure 4.4: The most simplistic value chain

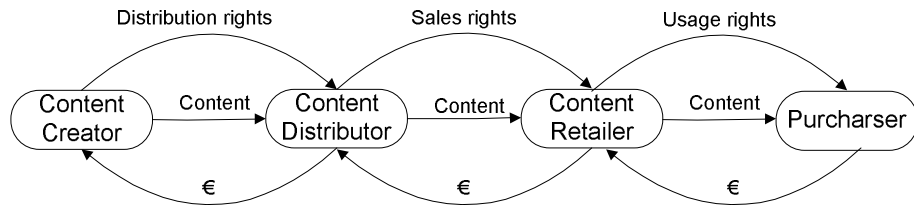


Figure 4.5: Value chain in an end to end scenario

`ec:contract-front` part, being the clauses under the `ec:body` element.

## 4.2 Value Chain Representations

One of the central ideas of this Thesis is that there is an intellectual property value chain for the multimedia contents, and that objects evolve and are traded in cascade, encompassing and enrolling different parties and different characteristic actions. The idea is actually not new and it has some precedents which are analyzed in this Section.

The most generic value chain would be that one placing an abstract category of intermediaries between the content creator and the content consumer, as in Fig. 4.4. As intermediaries, they can be placed here any kind of content transformers (editors, performers, sound engineers, metadata aggregators etc.) and content distributors (retailers, distributors, broadcasters, etc.). For example, [71] identifies the following activities: composing, performing, publishing, recording, reproduction, distribution and retailing. But for different content types (audio, video, etc.) and for different business models there are different steps. For example, focussing in the distribution, the value chain is declared as having only content distributor and retailer between content creators and purchasers [72] (Fig. 4.5), focusing in the content providers, a complex taxonomy is identified in [73].

The trends of the Web 2.0 where the importance of intermediaries and distributors is decreasing has also been repeatedly acknowledged [74] putting the traditional roles in difficulties [75].

Regarding the value chain representations, so far none has considered



relevant to give an explicit form excepting some isolated efforts. Thus, it is worth to mention de Rosnay's work [76], who in 2004 outlined an ontology for the value chain of music, although she didn't get into the details nor published the ontology; and even more akin, a copyright ontology [77] which also did not come about in practical applications.

The value chain in this Thesis is seen in terms of the Intellectual Property, and chains are identified as such if they are relevant according to the intellectual property model described in Chapter 2. This gives an accurate picture of what is in and what it is out of the chain. For example, a merchandising contract may be commercially very relevant, but does not attain to the intellectual property regards, and therefore it is disregarded in this Thesis. On the contrary, edition contracts, broadcasting contracts etc. are well studied under the auspices of intellectual property law and are the target of this Thesis.

Actually, the value chain from an intellectual property model happens to match fairly well with the parties treated in DRM systems, and the value chain model of this work matches it satisfactorily enough. For example, the contract management systems around the value chain described in [78] addresses the same parties that are addressed in this work, and they are not much different from those in [79].

But besides the existing business models, or the academic works in the literature, there are two particular model of the value chain which are of particular relevance and which are described more carefully here: the Functional Requirements for Bibliographic Records (FRBR) and the Conceptual Reference Model (CIDOC). The FRBR is a conceptual model widely accepted (but without a widespread ontologized version) which has had some echo in other ontologies (like the Music Ontology), and which describes the value chain in terms of an abstract multilevel representation. The Conceptual Reference Model is less related to this work, given that it focuses on cultural objects instead of general multimedia objects, but it has an accurate description and an associated ontology which are of maximum interest.

### 4.2.1 Functional Requirements for Bibliographic Records

The FRBR Functional Requirements for Bibliographic Records is a model designed in 1997 to restructure catalogue databases to reflect the conceptual structure of information resources. While it might appear far from the areas of affinity for this Thesis, it is of the highest interest given that its aim is to separate conceptually the different creation stages whose mapping to the intellectual property stages can be done immediately.

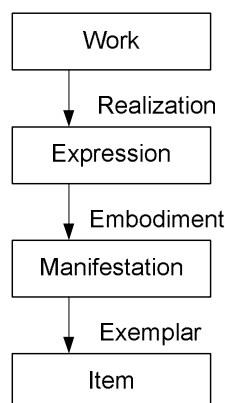


Figure 4.6: FRBR basic entities

FRBR uses an entity-relationship model of metadata for information objects, instead of the single flat record concept underlying in previous cataloging standards.

Entities in FRBR are categorised in three groups:

- Group 1: What is described by a record.
- Group 2: Who is responsible for the creation, production, etc. of the entity
- Group 3: Event, Object, Place, Concept

Records in FRBR refer to four levels of abstraction, comprising *work*, *expression*, *manifestation* and *item* (see Fig. 4.6). As an example, it can be seen how a video of “The Marriage of Figaro” may have different levels of annotations.

- *Work*: distinct intellectual or artistic creation.

The Marriage of Figaro, by Mozart

- *Expression*: the specific intellectual or artistic form that a work takes each time it is realized.

Figaro, by Carlo Maria Giulini, with Eberhard Wächter, Elisabeth Schwarzkopf, Anna Moffo, Giuseppe Taddei, Fiorenza Cossotto, Coro and Orquesta Philharmonia (Emi, 1960).

- *Manifestation*: the physical embodiment of an expression of a work.

Giulini's Figaro, released in Audio CD (1960)

Giulini's Figaro, released in Video VHS (1985)

Giulini's Figaro, released in Video DVD (2002)

- *Item*: a single exemplar of a manifestation

Giulini's Figaro, in Video DVD, item number 82323.

FRBR is widely used in libraries specially in the USA, and Europeana -the European digital Library, which manages 2 million records- is expected to have FRBR support by 2010. Besides this, it has influenced other initiatives. The next paragraphs exemplify it with a closer instance to the purposes of this Thesis: a music ontology.

Nothing is said about how the FRBR model is represented, although a RDF version has been proposed. The RDF version seems not to have gained much attention.

caption

```
@prefix frbr: <http://purl.org/vocab/frbr/core#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
<http://www.imdb.com/title/tt0245712/>
  a frbr:Work;
  dc:title "Casablanca"@es ;
  dc:title "Casablanca"@en ;
  frbr:realization <#casablanca> ;
<#casablanca>
  a rbib:MotionPicture ;
  dc:creator "Michael□Curtiz" ;
  frbr:embodiment <#amores-perros-dvd> .
<#casablanca-dvd>
  a rbib:DVD ;
  dc:date "2000"M
  frbr:item <#item2341>.
```

### A Semantic Web implementation: Music Ontology

The Music Ontology<sup>1</sup> (MO) is an attempt to link all the information about musical artists, albums and tracks in the Music world [80]. From its revision 1.11 in March 2007, the MO includes the music creation workflow. On the

<sup>1</sup>The Music Ontology specification is available at: <http://musicontology.com/>

contrary of FRBR, the Music Ontology gives an explicit OWL file as its normative form.

According to the FRBR model presented before, the MO includes the four key concepts of MusicalWork, MusicalExpression, MusicalManifestation and MusicalItem. But in the description of the workflow, other classes apart from those of FRBR are mentioned. MO defines a simple and a complex workflow:

- Simple Workflow. The simple workflow is the following.  
MusicalWork → Performance → Signal → MusicalManifestation
- Complex Workflow.  
Composition → MusicalWork → Performance → Sound → Signal → MusicManifestation

The steps in the workflow would be:

- Composition (optional): the event leading to the creation of a musical work.
- MusicalWork: the musical work itself.
- Performance: the event corresponding to an actual performance of the work.
- Sound: (optional) the physical sound produced by the performance.
- Recording (optional): The event representing the transduction from a physical sound to a signal, through the use of a microphone.
- Signal: recording the performance as a signal.
- MusicalManifestation: the release of this signal on a particular record.

The MO distinguish between events in the value chain (composition / arrangement / performance / recording) and objects (musical work, score, signal...).

The Music Ontology has been integrated in larger platform serving in fruitful applications, particularly powerful when interoperating with other ontologies [81].

### 4.2.2 Conceptual Reference Model and others

The CIDOC Conceptual Reference Model [82], is a formal ontology intended to facilitate the integration, mediation and interchange of heterogeneous cultural heritage information. The primary role of the CRM is to enable information exchange and integration between heterogeneous sources of cultural heritage information. Since 9/12/2006 it is official standard ISO 21127:2006.

More specifically, it defines and is restricted to the underlying semantics of database schemata and document structures used in cultural heritage and museum documentation in terms of a formal ontology. It does not define any of the terminology appearing typically as data in the respective data structures; however it foresees the characteristic relationships for its use. CIDOC CRM does not aim at proposing what cultural institutions should document; rather it explains the logic of what they actually currently document, and thereby enables semantic interoperability.

With respect to this Thesis it is of special interest how it defines some classes for objects (*legal objects*), rights, actors and the evolution of works (embracing creation, performance and production). It is also noteworthy the presentation of the ontology, not as a particular formal system, but with a narrative description of the classes (hierarchyzed), the relationships (with its cardinality restrictions) and their domain and range. The OWL ontology version<sup>2</sup> released in September 2009 is unofficial [83].

---

<sup>2</sup><http://purl.org/NET/cidoc-crm/core>



## Chapter 5

# Rights Expression Languages in DRM systems

For the regards of this Thesis, Digital Rights Managements are Information Systems able to control the access to digital assets. These digital assets represent objects of the intellectual property which are exchanged in a framework of security where the rights of the intervening parties are respected.

Within this framework, rights owners of the intellectual property objects express their conditions under which one or more rights over the object are transferred. In the most simple case, the owner of one musical work may sell the right of reproduction at a given price. But transfer of objects may not be limited to end products addressed to end users, as it is the case of the most known DRM systems. Digital Rights Management systems can be used between business partners and many other scenarios can be devised where rights are to be transferred. In this situation, agreed licenses (represented in one Right Expression Language or REL) truly are the digital counterpart of the contracts reviewed in Chapter 4.

### 5.1 DRM Systems

With the digitalization of artistic assets (and in general those goods subject to protection by the intellectual property laws), it appeared the idea of managing them in a digital environment and offering them for sale electronically.

The ease for reproducing the digital material led to the implementation of protection measures in order to refrain the massive distribution of protected material through parallel channels. In exchange of money, it was not offered a file in clear to download, or a stream easy to be captured. Instead, material was protected and delivered through secure channels, and new complex managing mechanisms appeared in what today is called Digital Rights Management (DRM) systems.

These ideas were manifested in the anti-copy mechanisms in DVD distribution, in the protected media formats etc., arousing general disapproval among the consumers. The common opinion of DRM as limiting systems (DRM was referred as digital *restrictions* management) came from the fact that they were initially conceived limiting too much the movements of the users. Compared to the limits imposed by the copyright law there happened a radical change: public law allows everything that is not explicitly forbidden, while DRM systems forbid everything that is not explicitly allowed.

Other reason for DRM's bad fame (and ultimately failure) is that it was only generally known a small fraction of the DRM systems, that affecting the end user<sup>1</sup>. But DRM systems can cover a wider gap in the trade process. The way Digital Rights Management systems is seen in this Thesis is broader, and it includes transactions between industrial partners and in general any kind of B2B commerce of intellectual property objects. Moreover, the view of DRM as a protection mechanism is superseded by a wider approach where rights are managed without necessarily being protected.

The next subsection will enumerate the most relevant initiatives on DRM excepting the MPEG-21 one, which is described with more detail in Section 5.2. Other abandoned systems have been omitted, like OpenMG, the DRM system used by Sony in its musical store until it was abandoned in August 2007 etc.

### 5.1.1 Overview of some DRM systems

This section merely enumerates some DRM systems along with their most prominent features. A rough division can be made in DRM platforms between those that are closed, proprietary solutions and those that intend to be interoperable, and which often are standard based and open source.

The later approach, defending that DRM systems is likely to benefit all

---

<sup>1</sup>In words of the authoritative voice of Renato Ianella: [...] *most commercial DRM systems have tended to focus on rights enforcement at the end of the value chain. The primary driver has been simply that the business case has been at the consumer end of the transaction. Hence, the enforcement "stigma" that DRM has had in the market place.* [84]



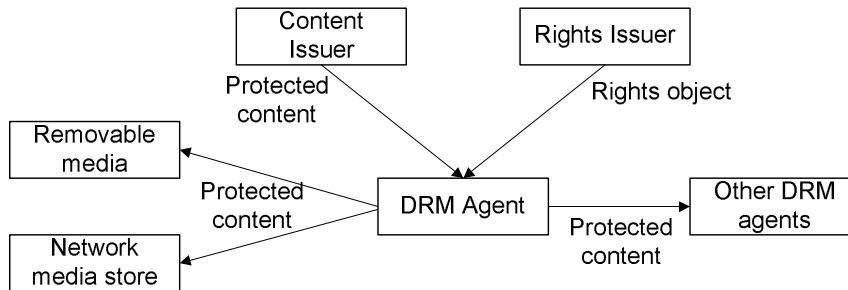


Figure 5.1: OMA DRM Architecture

the actors in the content value chain was already anticipated in [85], but the main advocates are the MPEG community, the DReaM initiative and the academia. In the other side, those defending their business model with a closed DRM platform are the rest of the industry giants like Microsoft, RealNetworks or Apple.

### Open Mobile Alliance DRM

This DRM system was invented by the Open Mobile Alliance (OMA), a forum which gathers most of the mobile phone manufacturers, operators and some IT companies. Its DRM system therefore represents the interests of most of the value chain, and is the most akin to the purposes of this work (apart from the MPEG-21 system reviewed in the next pages). OMA DRM is very important also for its widespread and for its neat definition, which can be found in [86].

OMA DRM spans quite well the value chain defining a set of actors and components in its reference architecture including the DRM agent, a content issuer, a rights issuer, a general user, and off-device storage. This view partially matches the concept of value chain, central for this Thesis, as it has been pointed out in [87]. Fig. 5.1 depicts how content issuers and rights issuers can be different actors communicating through a terminal (DRM agent) which in turn can communicate with other DRM agents or store the content and licenses. Content and licenses travel as different objects.

### Apple Fair Play

FairPlay is the DRM system of Apple. It worked delivering protected content from iTunes store to the device (iPhone, iPod, Apple TV, iPad etc.).

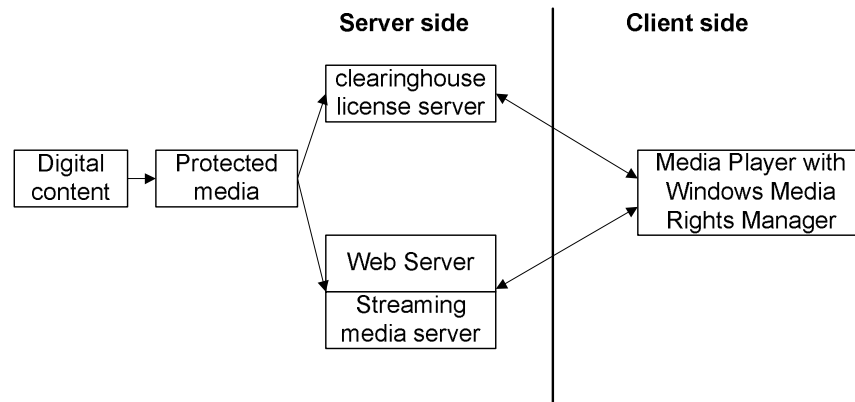


Figure 5.2: Windows Media DRM Architecture

FairPlay encrypted the content with a different key for each user, and the decrypting key was delivered and kept in a secure key store in the client device along with the account information. This schema was repeatedly broken but achieved a large commercial success. The file encryption was finally discontinued in March 2009.

### Windows Media DRM

Windows Media DRM is the Microsoft DRM system. It was also designed to provide secure delivery of audio and/or video content over an IP network to a PC in such a way that the distributor could control how that content is used. The key to decrypt the content is stored in an encrypted license, which is distributed separately (see Fig. 5.2).

On the contrary to FairPlay, Windows Media DRM was designed to be renewable, that is, it was designed on the assumption that it would be cracked and had to be constantly updated by Microsoft, achieving thus a relative robustness: it has usually not remained cracked for long. Microsoft offers six different APIs to help developers creating client-side applications that play back packaged Windows Media files.

### Helix DRM

Helix DRM<sup>2</sup> is also a proprietary DRM (developed by RealNetworks) aimed initially at protecting a particular content format (RealAudio and RealVi-

<sup>2</sup>Helix DRM, <http://www.realnetworks.com/products-services/helix/>

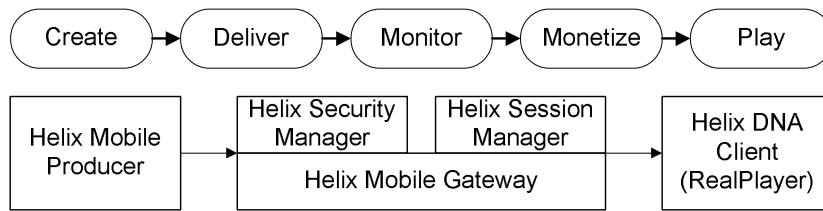


Figure 5.3: Helix DRM Delivery Platform

deo) but later also capable of protecting MP3, MPEG-4 Video, AAC and others formats. This DRM system is also intended to deliver the content to PCs and also non-PC devices like mobile and home appliances (see Fig. 5.3).

### OeBF DRM

Open eBook Forum (OeBF) is the trade and standards group for eBook industry, whose Rights and Rules Working Group aimed at creating a standard for interoperability of digital rights management (DRM) systems for the trusted exchange of electronic publications among rights holders. Their system as well as their rights expression language was specified [88] but no further developments have been published.

### AXMEDIS

AXMEDIS<sup>3</sup> was a research project partially supported by the European Commission under the Information Society Technologies programme of the Sixth Framework Programme (FP6), whose main outcome was a DRM platform for the content management and distribution.

The AXMEDIS platform included tools for the production, elaboration, management and protection of content (of different nature: audio, video, pdf, text etc.), as well as real multi-channel distribution (network, PDA, kiosk, mobile, i-TV etc..) in a DRM environment.

AXMEDIS designed and implemented a DRM architecture that consisted of several independent modules interacting as web services. Some of

<sup>3</sup>AXMEDIS (Automating Production of Cross Media Content for Multi-channel Distribution). IST 2004 511299. See AXMEDIS Products and Services Portal, <http://www.-axmedis.org>

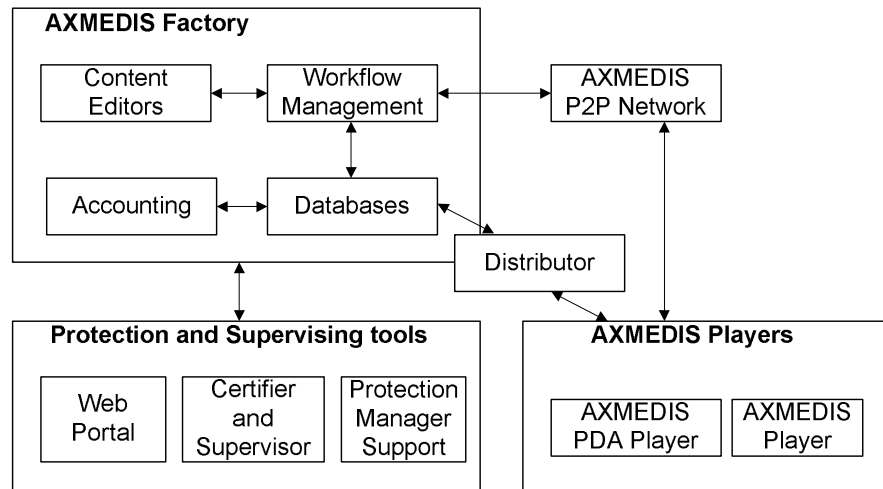


Figure 5.4: AXMEDIS Platform

the results of this Thesis have been incorporated in the AXMEDIS platform, whose main contribution is described in Section 8.2. AXMEDIS applications rested over an AXMEDIS platform or middleware, which had as most prominent components a Protection Processor responsible of verifying tools integrity and unprotecting multimedia content, a Protection Manager Support Client and Server, and a Certifier and Supervisor (AXCS) as the the authority to register and certify users and tools. The Protection Manager Support Client intended to manage and store protection information, licenses, reports and other secured information in a local secure storage system, being also responsible for authorizing users to perform actions over objects and of delivering protection information. The Server side was intended to create and store rights expressions, adapts rights expressions and perform authorization of content usage based on licenses (see Fig. 5.4).

### OpenSDRM

Open and Secure DRM (OpenSDRM [89]) is a DRM platform with a public specification and an open-source implementation, born from an European FP5 Information Society Technologies project called Moses<sup>4</sup>. OpenSDRM relied on a distributed-functionality paradigm where each of the DRM com-

<sup>4</sup>MPEG Open Security for Embedded Systems (MOSES), at <http://atlantis.tilab.com/-projects/moses/index.htm>

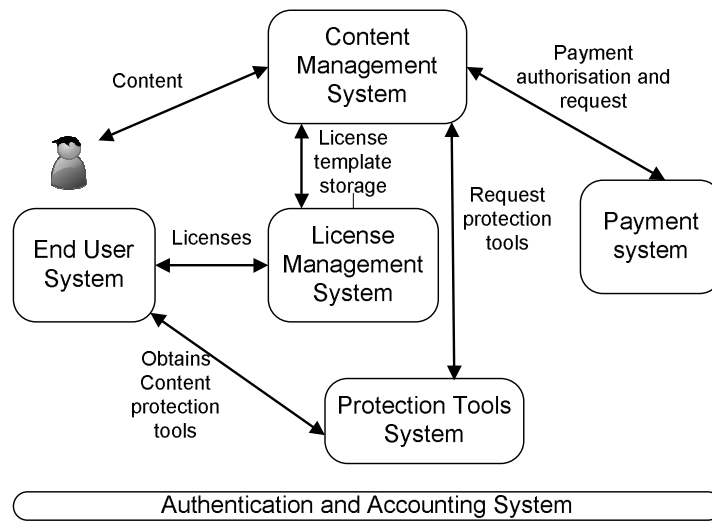


Figure 5.5: OpenSDRM architecture

ponents was a Web service, and all messages between the components were based on SOAP over SSL channels. This and other open systems are reviewed in [90] where an analysis of the various methods for implementing interoperable digital rights management platforms is given.

Fig. 5.5 shows the main elements in the OpenSDRM architecture, where the user is acting as both content provider and content consumer. OpenSDRM is composed by a set of external elements and a set of internal components exchanging SOAP messages. The protection tools system represents any organization providing tools for encrypting, scrambling or watermarking the content which has adopted the OpenSDRM messages protocol. Commerce platforms and content management systems (the upper box in Fig. 5.5) can also be external systems but adhering to the OpenSDRM interfaces, as well as the payment infrastructure. The authentication system is internal to the OpenSDRM platform and is responsible of authenticating both the internal and the external agents in the architecture. The license manager (in the center of Fig. 5.5) provides the internal service of accepting, storing and validating licenses which can be either MPEG-21 REL or OMA DRM. Being an open initiative, OpenSDRM started formally a project hosted in SourceForge<sup>5</sup>, but unfortunately no code has been uploaded so far.

<sup>5</sup>OpenSDRM <http://sourceforge.net/projects/opensdrm/>

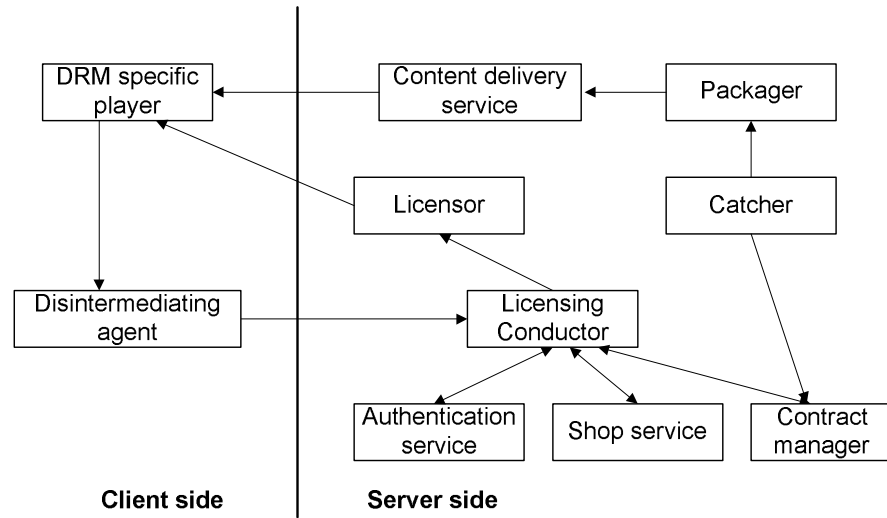


Figure 5.6: DReaM Architecture

## DReaM

The Project DReaM is the Sun open source project to develop their Digital Rights Management (DRM) solution [91], also intended for different kinds of content and multiple devices. The main idea of DReaM is chasing interoperability between different systems, declaring already in the specifications that whenever an adequate proprietary solution appears, DReaM will be capable of integrating with these solutions providing openness and interoperability. The DReaM architecture supports the separation between the rights management components through the decoupling of authentication (identification in DReaM is not made at a device level but at a network identity allowing different identities per device), licensing, rights management and protection systems. This disintermediation enables the choice and selection of these technologies independent of each other without any compromise for the overall solution.

This intermediate protocol is called DReaM-MMI and may be used for direct interaction between any DReaM-MMI enabled client and a license server. DReaM architecture is depicted in Fig. 5.6.

### Marlin

The aim of Marlin is similar to that of DReaM, aiming at creating a DRM system able to interoperate among devices from different vendors. Marlin is an evolution of the NEMO and Octopus (a toolkit for building DRM engines). NEMO (Network Environment for Media Orchestration) is a secure messaging architecture based on web services for digital media distribution and rights management and it is based on X.509 identity certificate and SAML security assertion standards.

Sony uses Marlin in the PlayStation Network, and it has also commercially deployed as the basis of the national IPTV standard in Japan<sup>6</sup>.

### Coral

Another consortium which has issued a DRM towards interoperability has been Coral, an international, cross-industry consortium which has defined a NEMO-based system [92]. More precisely, Coral itself is not a DRM system, but it provides the means to interoperate with different DRM systems, by defining the abstract services. The philosophy is to standardize a framework for DRM interoperability, not the DRM system itself.

Coral's architecture define a set of roles (functions) and nodes (devices and applications). These roles also span the value chain and include a DRM Content Exporter, the Content Mediator, the Rights Mediator, DRM Content Importer, and the DRM License Mapper (this one mapping rights in one system to rights in another by making reference to a set of policies).

#### 5.1.2 Common elements in DRM systems

A DRM system is thus a complex system that is not homogeneously understood by the different implementors, and each DRM system has its particularities. Nevertheless, some of these elements appear recurrently through the different systems and these features can be described here.

#### Content creation

The first step in the chain -and without which there is no chain- is the creation of digital objects. These digital objects usually have a precise format, and the step of creation may actually be data injection from other existing databases. Accompanying the digital objects, usually metadata is present so that searches can be possible as well as an easier management. Digital

---

<sup>6</sup>[http://www.marlin-community.com/overview/success\\_stories/actvila](http://www.marlin-community.com/overview/success_stories/actvila)

objects may be protected so that they travel safe and even though they are reached by a malicious party, data is still safe.

One example of content creation tools in the different platforms is the Helix DNA Producer application of Helix DRM. The Microsoft equivalent is the Windows Media Encoder, which comes along an API to create protected content. Microsoft's DRM core is the Windows Media Rights Manager, which packages the digital media file. The Marlin product to create content is called Bento4 Packager, and the AXMEDIS equivalent is the AXMEDIS Editor.

### **Content distribution**

Rights are managed in DRM because content is delivered somewhere. This delivery is called distribution, and there are of course many distribution networks (mobile, Internet, private networks) and means (broadcast, download, streaming etc.). If the content is protected, then it is not only needed to distribute the content itself but also the accompanying decryption keys for the authorized consumer.

The Helix DNA Server can stream media files over a network in Helix DRM, but is not the only Helix-based platform (the most famous is the Rhapsody online store). Given that Microsoft distributes separately the content and the key, the content provider has to set up a content distribution service plus a Windows Media Rights Manager License Service. Marlin software to deliver content is called Bluewhale Marlin Broadband Server.

### **Content consumption**

Content is consumed in a variety of ways too. There can be DRM player specific for the DRM system, or there may be standard applications. The application used to reproduce or render the content is generally known as *player*.

Ad hoc applications are able to take the needed measures to decrypt the content while being safe against intruders. If the user is authorized content is deciphered and consumed. Typical DRM players have secure storage, execution and consumption environments where unauthorized parties cannot access keys and status information.

While the only FairPlay's DRM players have been iTunes, iPods and other Apple's products, the other companies offered creating different reproducers and so besides Windows Media Player, Microsoft offers a common API for PC and other devices (like Zune) etc. Helix has also its own DRM



Helix DNA Player (core of RealNetworks applications) as open source and free for new implementations. Marlin also offers an API to create client-side DRM applications (Sushi Marlin Client SDK).

### Report of Events

The fact of management implies usually having a precise knowledge of what is going on in the DRM system. For this, the most important events are notified to an event receiver who will operate in consequence (billing, taking measures, writing reports etc.). Besides the DRM system administrator, some other participants of the value chain may be interested in some events, e.g. an author would like to be informed about the sales so that the royalties are properly calculated. Therefore, a mechanism is necessary to allow systems to share information about events referred to multimedia content and peers that interact with the content, this mechanism is usually called *event reporting* [93].

### Security in the DRM systems

All the modules and components described in the framework of a DRM platform must implement certain security measures for the system to work safely. A review of the security measures in DRM systems can be found in [94] and [95].

## 5.2 MPEG-21 as a Framework for DRM Systems

### 5.2.1 Overview of MPEG-21

MPEG (Moving Picture Experts Group) is a working group aimed at developing standards for coded representation of digital audio and video and related data. It is formally the Joint Technical Committee for Information Technology (ISO/IEC JTC1 SC29) between ISO and IEC. ISO is the International Organisation for Standardization and IEC is the International Electrotechnical Commission, two standardization bodies which join their efforts in this and other matters.

Since 1998, MPEG has produced standards exceeding their initial task. Besides video and audio coding standards (MPEG-1, MPEG-2, MPEG-4) and its transport, it has developed multimedia content description standards (MPEG-7) and ultimately a whole multimedia framework, the MPEG-21 set of standards. This MPEG-21 standard pays attention to the intellectual

property management and protection of the multimedia items and is the natural environment for the development of this Thesis.

According to its foundational document, MPEG-21 aims at defining a normative open framework for multimedia delivery and consumption for use by all the players in the delivery and consumption chain. This open framework tries to provide content creators, producers, distributors and service providers with equal opportunities in a fair environment. Put it in another way, *environment* can be here read as *DRM system* or framework for *DRM systems* given that it defines a controlled content exchange system or at least the foundations to build such a system.

MPEG-21 is based on two essential concepts: the definition of a fundamental unit of distribution and transaction and the concept of users interacting with these units. These units of abstract information are called Digital Items. Digital Items are XML-based documents constituted by content, metadata and structuring information. Thus, adjacent to the mere content (expressed in *resources* in this context), it can be found annotations, intellectual property management and protection information, rights expressions etc. All these specifications are given under the form of standard Parts.

The MPEG-21 standard is currently formed by eighteen Parts, plus the nineteenth which is the result of this Thesis. These Parts have been elaborated asynchronously and with an heterogeneous scope and complexity, what makes a somewhat cumbersome standard.

- Part 1: Vision, Technologies and Strategy. It describes the vision of the multimedia framework, and it also defines the strategy for achieving the standard objectives.
- Part 2: Digital Item Declaration (DID). It describes the Digital Item as the unit of information to be acted upon (managed, described, exchanged, etc.) within the model, and it describes the meta-language to define these Digital Items.
- Part 3: Digital Item Identification (DII). It provides the schema to be used to uniquely identify elements within and outside the Digital Item.
- Part 4: Intellectual Property Management and Protection (IPMP). It defines an interoperable framework for Intellectual Property Management and Protection. It does not define any protection mechanism, but describes how to interface with algorithm providers (IPMP tools). A good reference can be found in [96].

- Part 5: Rights Expression Language (REL). The REL license is a document expressing the rights that can be exercised over a Digital Item and the conditions for that exercise to be authorized. Along with the XML schemata to produce licenses, an authorization mechanism is described. This Part is described with detail in [97].
- Part 6: Rights Data Dictionary (RDD) [98]. The Rights Data Dictionary declares the terminology to be used within the REL licenses, in a hierarchical structure. It also refers to a registration authority in charge of keeping this dictionary up to date.
- Part 7: Digital Item Adaptation (DIA). Adapting Digital Items means transforming the Digital Items to the most adequate context conditions, like network traffic, rendering devices capabilities or user preferences. The DIA specifies also the needed XML schemata.
- Part 8: Reference Software. This miscellaneous ragbag includes software given as an aid to the reader. However, the lack of coordination in this Part and the different origin of the contributors makes this hodgepodge of little use.
- Part 9: File Format. This Part describes a standard file format to encode Digital Items, either as computer files or as streams.
- Part 10: Digital Item Processing (DIP). It provides a set of tools for specifying processing of a Digital Item in a predefined manner. Each Digital Item can describe itself the list of meaningful methods that can be applied to it, and that are to be applied by a DIP Engine.
- Part 11: Evaluation Methods for Persistent Association Technologies. This non-normative Part was aimed at evaluating persistent association technologies, like watermarking algorithms, to be evaluated.
- Part 12: Test Bed for MPEG-21 Resource Delivery. This Part provides a software-based test bed for the scalable delivery of media, under different conditions like varying network environments, different demanded qualities etc.
- Part 14: Conformance Testing. This well-intentioned Part was aimed at defining conformance testing for other parts of MPEG-21 but has been in practice largely ignored.

- Part 15: Event Reporting (ER). Events which happen around the digital items and the agents operating on them can be represented with the XML Schemata given in this Part, and handled as suggested by the ER architecture.
- Part 16: Binary Format. Binary formats allow the efficient encoding of Digital Items.
- Part 17: Fragment Identification of MPEG Resources. It defines a normative syntax for URI Fragment Identifiers.
- Part 18: Digital Item Streaming. Describes the tools for Digital Item Streaming, as well as how Digital Items are mapped to delivery channels like the MPEG-2 Transport Streams or the Real Time Protocol.
- Part 19: Media Value Chain Ontology (MVCO). Is the central part of this Thesis and described sufficiently in Chapter 6.

## 5.2.2 MXM as a MPEG-21 implementation

### MXM vision

The MPEG group constitutes a body which has produced a large amount of documents and standards. Many of them have been a breakthrough in their fields, specially those describing algorithms for video and audio encoding. However, regarding practical implementations, MPEG only provides unpretentious reference software to demonstrate the capabilities of its standards without concerning efficiency or practical integration in applications.

MXM (MPEG Extensible Middleware<sup>7</sup>) strives at overcoming these deficiencies and offers an integrated middleware for fast developing of multimedia applications, without needing to have in-depth knowledge of the underlying technologies, and with the advantage of a modular structure which enables replacing certain blocks or making use of existing ones.

### Origins of MXM

In the vision of interoperability and openness of DRM, it is also fundamental to have an open implementation demonstrating the posed capabilities. As it was said in Section 5.1, MPEG has advocated for interoperability and MXM is only the culmination of efforts which had started some years before [99]. The beginning can be dated back to late 2003, when Leonardo Chiariglione,

---

<sup>7</sup>MPEG Extensible Middleware <http://mxm.wg11.sc29.org>.

convenor of the MPEG meetings, drew up a Digital Media Manifesto [100]. The immediate consequence was the establishment of a group of experts called Digital Media Project which set out to make this idea true, reaching after several years the specification of a DRM Interoperable Platform (IDP), a platform technically open to different value-chain players and for heterogeneous end-user devices.

The IDP specification is given as a set of nine approved documents, including the requirements [101], the architecture [102], the platform specification [103], the use cases and value chains analysis [104], the certification authorities description [105], a terminology [106] and a reference software [107]. A good summary of all of it can be found in [108].

An implementation of the IDP specification was then given by the open source Chillout project<sup>8</sup>. Chillout is based on several MPEG standards - mostly MPEG-21 (ISO/IEC 21000) and MPEG-A (ISO/IEC 23000)-, and it was demonstrated during the Beijing Olympics (WIM TV trial<sup>9</sup>) in 2008. The Chillout project was the seed for the MXM project, once blessed by the MPEG community and officially baptized in MPEG as MPEG-M and in ISO/IEC as standard 23006. As of the editing of this Thesis, MXM was at an advanced step in its standardization process.

The MXM standard is organized in four public documents, describing the architecture [109], the APIs [110], the reference software [111] and the protocols [112]. The latter document had a previous version as standard ISO/IEC 29116 [113].

### MXM Architecture Elements

MXM defines an architecture comprising MXM APIs (*MXM engines*) on top of which *MXM applications* run on *MXM devices*, as it can be seen in Fig. 5.7. The MXM Device is described as a platform able to run one or more MXM Applications; the applications run on top of the Operating System, and additionally may exploit a number of MXM Engines enabling access to the MXM technologies and protocols.

Each MXM Engine is backed by a MXM API (defined either in Java, C++ or both), and the interoperation between MXM Engines is well defined through the protocols. Protocols exchange a well defined set of XML messages, in practice delivered through Web Services.

The MXM standard is mainly concerned with the interfaces of the MXM Engines, named MXM Engine APIs and the interface to a master Engine

---

<sup>8</sup>The Chillout Project (2009) <http://chillout.dmpf.org>

<sup>9</sup>See <http://www.wimtv.it/>

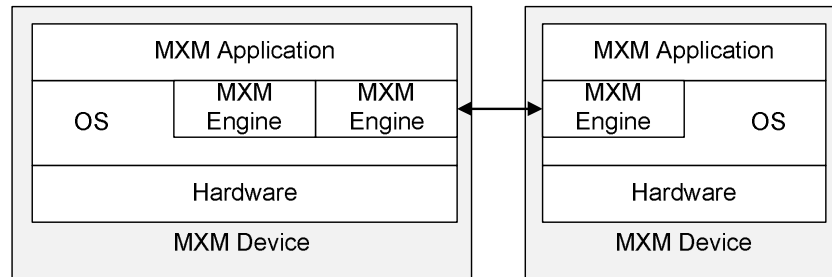


Figure 5.7: MXM Components model

called Orchestrator Engine. The specification also defines the API calls, but not how they may or should be implemented. The implementation may actually be different for different computing platforms (either hardware or software, like the OS). It is likely, however, that a sample model implementation is given as a Reference Software by the standard itself. In the current specification, MXM defines 18 engines, which are listed here:

- *Orchestrator Engine*, a special engine capable of invoking predefined sequences of calls in the others engines and capable of providing a simpler, unified interface to a priori known application domains.
- *Media Framework Engine*, supporting video, audio, image and 3D graphics handling, it provides methods to create (encode) and access (decode) the elementary streams.
- *Rendering Engine*, providing the access to hardware graphics acceleration, input device handling and its abstraction.
- *Metadata Engine*, for creating and accessing audio, still image, 3D graphics and video metadata.
- *Content Search Engine*, implementing the MPEG Query Format [114] methods to access content repositories (the content provider device).
- *Digital Item (DI) Engine*, concerned with the Digital Item Declaration, the Digital Item Identification and all related operations to create access and edit Digital Items.
- *Content Protocol Engine*, with the procedures to identify a content item from a content identification device, to access it and store it

from/to a content provider device and to authenticate it completely or in parts.

- *MPEG-21 File Engine*, concerned with the serialization and access of Digital Item as files (with the ISO/IEC 21000-9 MPEG-21 File Format).
- *Digital Item Streaming (DIS) Engine*, concerned with the serialization and access of Digital Item as streams, operating over ISO/IEC 21000-18 Digital Item Streaming data structures [115].
- *Digital Item Adaptation (DIA) Engine*, providing the proper methods to parse, access and create information contained in usage environment description elements (context information needed to perform content adaptation).
- *Event Reporting (ER) Engine*, to track every relevant action in the system, as well as to create Event Report Request and Event Reports conforming to ISO/IEC 21000-15 [116].
- *Security Engine*, providing cryptographic algorithms, digital signature and the tools to achieve trust in devices. Note that devices interoperate with each other after having mutually authenticated themselves, by exchanging certificates provided by the Certification Authority.
- *Media Value Chain Ontology (MVCO) Engine*, with the methods to access the value chain ontology and manage Digital Items and users in conformance with the IP value chain model.
- *Rights Expression Language (REL) Engine*, capable of parsing and editing MPEG-21 REL licenses, according to the standard and its extension with profiles.
- *License Protocol Engine*, declaring the methods to access store and revoke licenses remotely in a license provider device.
- *Domain Engine*, implementing the methods to exchange information with the domain management device.
- *Intellectual Property Management and Protection (IPMP) Engine*, providing classes to create and access IPMP data structures, which describes the mechanisms and tools for the protection of Digital Items.

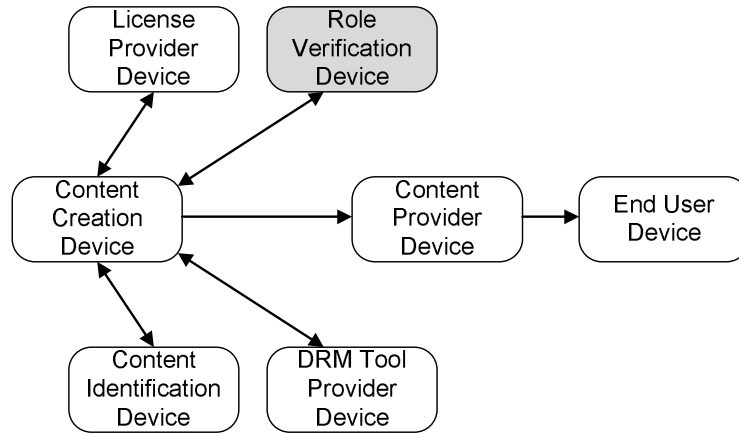


Figure 5.8: MXM Architecture

- *IPMP Tool Protocol Engine*, providing the methods to access an IPMP Tool Body, i.e., how to access an implementation of protection algorithms.

MXM Engine APIs are divided in four categories: creation APIs (including APIs to create data structures, files, elementary streams, etc.), edition APIs (APIs to modify an existing data structure, file, elementary stream, etc.), access APIs (APIs to parse data structures, files, decode elementary streams, etc.) and presentation APIs (APIs to render or show to the user the content of data structure, files, elementary streams, etc.)

Fig. 5.8 displays some of the devices cooperating together in a possible architecture. The *content creation device* is the tool aimed at creating content. But this content needs to be uniquely identified (hence the arrow to the *content identification device* which will give it a unique identifier, needs to be licensed (hence the arrow to the *license provider device*, which will associate the content with a license, needs to be protected (hence the arrow to the *DRM tool provider device* which will encrypt the content) and needs to be related to the work where it derives from (hence the arrow to the *Role Verification Device*, which will keep a track of the derivative relationships). The latter is shadowed because it is the result of the work presented in this Thesis. The created content is finally passed to the content provider device, which in turn will sell it (or freely distribute) and deliver it to the end user device to be reproduced.



## 5.3 Right Expression Languages

License is the document written in a Rights Expression Language (REL) as a part of a Digital Rights Management system conveying an authorization to perform certain actions. This Section reviews the different ways in which licenses have been expressed.

To date, some different RELs have been proposed with a different degree of success or deployment; for the interest of this Thesis MPEG-21 REL will be described more in-depth.

### 5.3.1 MPEG 21 REL

#### Origin

The development of today's rights expression languages started with the work of a Xerox PARC's researcher called Mark Stefik. Stefik's work began in the early 1990's with a statement of the need for protection for digital materials in order to foster online commerce. As part of that system he needed to develop a machine-readable vocabulary to express rights in a trusted software system, and so he started working on the Digital Property Rights Language. DPRL was patented by Xerox in November 1994 [117], and four years later the first XML version was launched as DPRL 2.0, replacing the LISP format of DPRL v1.0.

XrML, the XML language to express rights, was licensed to company founded by Microsoft and Xerox, ContentGuard, which evolved DPRL into the eXtensible Rights Markup Language (XrML) [118]. Version 1 of XrML was published in 2001. It was able to represent the issuer of the license and the grants he issued. In term, these grants consisted of the principal (person who enjoys the granted rights), the granted rights, the resource over which the rights applied, and the conditions under which they could be exercised.

In a new version, it included features describing some methods to make the REL robust, such as unique identifiers, private and public keys, and other mechanisms for identifying and verifying the authenticity of the issuer and the user of the resource, like certification for hardware and software that would be part of a trusted environment. The rights list remained the same but with new definitions, distinguishing clearly between those rights that created a new resource versus those that modified an existing resource.

Version 2 of XrML was published in 2002 and broke the previous line of development. XrML was made more abstract, able to represent any kind of media in any kind of situation. Therefore, the list of rights was modified, and concrete terms disappeared, remaining only those needed to establish

reliable frameworks where to introduce the particularities. This version might have been academically sound, but proved to be too complex. In 2003, XrML was used as the basis for the rights expression language for the MPEG-21 standard, and its basic structure lasted.

### MPEG-21 REL language elements

On April 2004, MPEG-21 Part 5 was published as an Standard of the International Standard Office (ISO), the ISO/IEC 21000-5. MPEG-21 REL defined REL as *an XML-based language for expressing rights related to the use and distribution of digital content as well as access to services*.

Soon after, a complementary standard was approved as ISO Standard, the RDD (Rights Data Dictionary). This part *describes a Rights Data Dictionary which comprises a set of clear, consistent, structured, integrated and uniquely identified terms to support the MPEG-21 Rights Expression Language (REL)*. This shall be discussed later on.

The MPEG-21 REL standard is very well described by its XML representation under the form of XML schemata. The root element in the schema is the *license* element, given that the main objective of MPEG-21 REL is representing licenses.

The elements in a license are shown in Fig. 5.9. The schema actually spans in three XSD files (core, standard extension and multimedia extension), and these XML Schemata are normative. A REL license consists of an *issuer* element and a set of *grants*, eventually organized in *grant groups*. Each license is identified by a license id (an attribute of the license element) and an optional *title*, intended for human consumption rather than automatic processing. Licenses can carry also an inventory for defining elements to be reused in the grants, reducing thus redundancy and verbosity in licenses. Finally, licenses can also have an element with undefined content for its extension (the element *other info*). The license can come eventually encrypted.

In this schema, the issuer is the user who emits one or more *grants*. The grant is the permission given to another issuer (the *principal*) to execute a certain action (the *right*) over a certain *resource* subject to certain *conditions*.

The principal element appears only once in a grant, but it can represent several actual users, through the use of the `r:allPrincipals` element, a logical conjunct of the principals represented by all of its children. It can also represent a generic user in possession of a private key adequate for a given public key (`r:keyHolder` element).

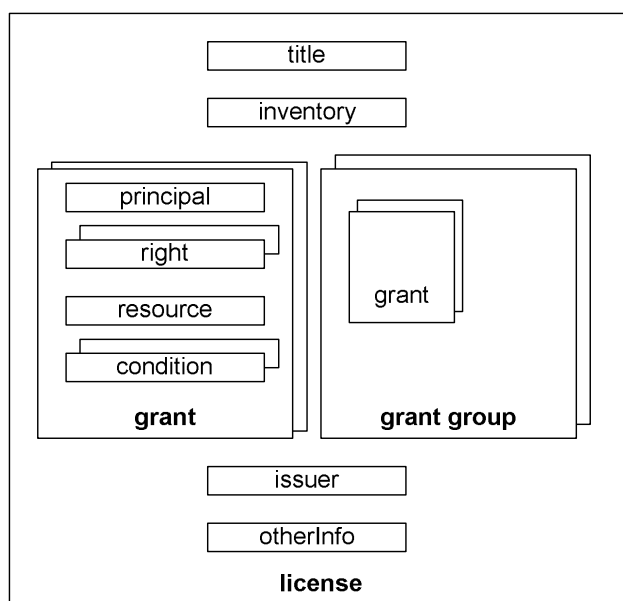


Figure 5.9: REL license

Rights are vaguely related to those declared in the RRD part, which accounts for a hierarchy of rights textually defined and for an extension and registry mechanism with the theoretical existence of a RDD registry.

The MPEG-21 REL has been extended with different profiles. A profile is a subset of the REL oriented for a specific goal (for example broadcasting) but defining eventually new elements. The currently defined profiles are:

- MAM (Mobile And optical Media) profile [119]
- DAC (Dissemination And Capture) profile [120]
- OAC (Open Access Content) profile [121]

These profiles take some of the existing elements in REL and define some new others (i.e. see the rights in Fig. 5.3 where *m1x*, *m2x* and *m3x* are namespaces for the new elements).

### MPEG-21 REL rights

The MPEG-21 REL standard defines three XML schemata, namely the core schema (usually prefix *r* for this namespace), the standard extension (names-

<i>Right</i>	<i>Meaning</i>
<code>r:issue</code>	to issue a license on the referred resource
<code>r:obtain</code>	to have access to a rights offer
<code>r:possessProperty</code>	to be attributed a certain property
<code>r:revoke</code>	to allow revoking an existing license

Table 5.1: MPEG-21 core rights

pace *sx*) and the multimedia extension (*mx*).

The core schema of MPEG-21 REL declares only a few general constructs to express rights (see Table 5.1), but of structural importance: they allow creating end user licenses and re-distributor licenses (`r:issue`), they define the mechanisms to revoke licenses (`r:revoke`), or to make rights offers etc.

The multimedia rights are those applied on general multimedia resources; they are listed in Table 5.2. Together with the right element, these tables show an informal definition, present at earlier stages of the standard but clearer than in their final form. The meaning was supposed to be completed with that given in the Rights Data Dictionary (RDD), but the standard actually does not clarify much nor has an updated account with the new rights (for example with those of the REL extensions).

New rights defined in the MPEG-21 REL profiles are listed in Table 5.3, but they cannot be arbitrarily used. Adhering to a REL profile implies limiting the palette of available rights. Thus, the OAC profile only accepts one of these rights: `mx:adapt`, `mx:execute`, `mx:play`, `mx:print`, `m1x:governedCopy` and `m3x:governedAdapt` but no others.

### MPEG-21 Rights Data Dictionary (RDD)

The RDD was conceived to facilitate the accurate exchange and processing of information between interested parties involved in the administration of rights, and it tried to precise the meaning for the terms defined in the REL.

In RDD, terms are given a definition, a structured hierarchy and a set of attributes. The standard adopts the form of an ontology not formally specified but with narrative text, and it is open for its extensions. An authority is supposed to be appointed to centralize these extensions (the Registration Authority), and it works under the premises of payment per registered term.

Soon after RDD was approved as Part 6 of MPEG-21, a bitter controversy started around the intrinsic contradictions in the document, unavoidable for the implementations based on the given text, and a series of

<i>Right</i>	<i>Meaning</i>
<code>mx:modify</code>	to change a resource, preserving the alterations made
<code>mx:enlarge</code>	to modify a resource by adding to it
<code>mx:reduce</code>	to modify a resource by taking away from it
<code>mx:move</code>	to relocate a resource from one place to another
<code>mx:adapt</code>	to change transiently an existing resource to derive a new resource
<code>mx:extract</code>	to take a part out of an existing resource to derive a new resource
<code>mx:embed</code>	to put a resource into another resource
<code>mx:play</code>	to derive a transient and perceivable representation of a resource
<code>mx:print</code>	to make a fixed physical representation, such as hard-copy prints of images or text, that may be perceived (without any intermediary process) with one or more of the five human senses
<code>mx:execute</code>	to execute a digital resource
<code>mx:install</code>	to follow the instructions provided by an installing resource
<code>mx:uninstall</code>	to follow the instructions provided by an uninstalling resource
<code>mx:delete</code>	to destroy a digital resource

Table 5.2: MPEG-21 multimedia extension rights

<i>Right</i>	<i>Meaning</i>
<code>m1x:governedCopy</code>	to copy the resource and at the same time result in certain rights being associated to the copied resource
<code>m1x:governedMove</code>	to move the resource and at the same time result in certain rights being associated to the moved resource
<code>m1x:enlist</code>	to link the related resource into a new playback control sequences description (i.e. playlist) for optical disc
<code>m1x:delist</code>	to unlink or delist (the reference to) the related resource from a related playback control sequences description (i.e. play-list) for the optical disc when the play-list is newly created from an existing one
<code>m2x:export</code>	to export the associated broadcast program to another rendering or storage
<code>m2x:extendRights</code>	to extend the rights which are the originally transmitted
<code>m3x:governedAdapt</code>	to adapt the resource and results in certain rights being associated with the adapted resource

Table 5.3: MPEG-21 rights in the profiles

amendments were made to the standard. No actual use of RDD has been publicly released.

### MPEG-21 REL Authorization

Software implementations making an authorization decision using REL licenses should follow the guidelines of the so called REL Authorization Model. Authorizations permit a user to exercise a right against a resource. The decision makes use of an authorization request, an authorization context, an authorization story, and an authorizer. An authorization request represents the question whether it is permitted for a given principal to perform a given Right upon a given resource during a given time interval based on a given authorization context, a given set of licenses, and a given trust root. The authorization story contains the grant to be authorized, the grant that demonstrates its validity and an authorizer.

Globally, the authorization process is a not well defined operation, depends on the actual implementation and complete implementations have never done or at least made public.

### 5.3.2 ODRL

ODRL (Open Digital Rights Language [122]) is also an XML-based standard Rights Expression Language (REL), conceived in 1997 by John S. Erickson and Renato Iannello. It was proposed in 2000 as an open standard, and nowadays is in use in various applications in Australia and Europe, primarily in academic and digital library environments. Its most important commercial application is in the wireless message protocols for mobile devices, in development by the Open Mobile Alliance (OMA), Open Mobile Alliance.

ODRL is managed by an organization open to public participation. It has created a profile that supports Creative Commons licenses and is working on a profile for geospatial data and a profile for Dublin Core Metadata Initiative (DCMI) metadata. There is at least one open source implementation of ODRL available. At the editing time of this document, ODRL had published a draft of their second version<sup>10</sup>.

The ODRL model consists of three core entities: *assets*, *rights* and *parties*, and with these three core entities, both offers and agreements can be expressed.

---

<sup>10</sup>ODRL version 2.0 - Core Model Draft Specification, <http://odrl.net/2.0/DS-ODRL-Model.html>

The assets include any physical or digital content, and they are equivalent to MPEG-21 resource element in the license schema. Assets can refer also to intangible objects, given that ODRL acknowledges the IP entities model of FRBR (see Section 4.2.1).

The rights (akin to MPEG-21 grants) include permissions (MPEG-21 rights) which can then contain constraints, requirements, and conditions. Constraints can be imposed relating to the user, the device, the bounds (spatial etc.), the time, the aspect or the target. Requirements can be set regarding a fee, some needed interactions (like 'the user must register') or usage.

The parties (MPEG-21 users) include end users (MPEG-21 principals) and rights holders (MPEG-21 issuers).

### **Rights in ODRL**

Rights (in ODRL are called permissions) can be used for both agreements and offers, and they are classified as usage permissions (pertaining to the end use of an asset), transfer permissions (pertaining to the downstream transfer rights of an asset), asset management permissions (pertaining to the digital management of an asset) or reuse permissions (pertaining to the re-use of an asset creating a new asset). The ODRL Data Dictionary for permissions is shown in Fig. 5.4.

### **5.3.3 Other REL languages**

#### **METSRights**

METSRights is an extension schema to the popular METS (Metadata Encoding and Transmission Standard) packaging metadata standard [123].

METS is a XML standard for encoding descriptive, administrative, and structural metadata regarding objects within a digital library. It is maintained in the Network Development and MARC Standards Office of the Library of Congress of the USA, so it has a main use for books and printed publications. Nevertheless, it can be used for musical scores, photographs, compact discs, recorded events etc. too. Many projects using METS are dealing with materials that are primarily archival in nature, owned by a single institution but available to the greater research community.

One of its extensions, is the METSRights schema<sup>11</sup>, which can express the rights associated to all of these digital library materials. It is divided into

---

<sup>11</sup>The XML Schema can be found at <http://www.loc.gov/standards/rights/METS-Rights.xsd>

<i>Permiss.</i>	<i>Definition</i>
Usage Permissions	
<b>display</b>	rendering the asset onto a visual device
<b>print</b>	rendering the asset onto paper or hard copy form
<b>play</b>	rendering the asset into audio/video form
<b>execute</b>	executing the asset
Transfer Permissions	
<b>sell</b>	allowing the asset to be sold (ownership transfer) in exchange of value
<b>lend</b>	allowing the asset to be made available for temporary the lende use then returned. During the period, asset is only available to the lende
<b>give</b>	allowing the asset to be given away (ownership transfer) in perpetuity without exchange of value
<b>lease</b>	allowing the asset to be made available for a period of time then returned. During this period, the asset is only available to the lessee
Asset Management Permissions	
<b>move</b>	allowing a digital asset to move between data storage devices
<b>duplicate</b>	making an exact copy of a digital asset between data storage devices
<b>delete</b>	deleting a copy of an asset.
<b>verify</b>	allowing authorization to check the authenticity of an asset
<b>backup</b>	making copies of an asset for the purpose of guarding against the loss of the original due to accident or equipment failure
<b>restore</b>	allowing the conversion of a backup copy into a usable copy in a controlled manner.
<b>save</b>	saving a copy (including any changes) of an asset to permanent storage.
<b>install</b>	allowing for the operation of loading, verification and certification of an asset into a data storage device.
<b>uninstall</b>	allowing for the removal from or disabling of an asset in a data storage device.
Reuse Permissions	
<b>modify</b>	changing parts of the asset creating a new asset
<b>excerpt</b>	extracting (replicating) unchanged parts (or all) of the asset for reuse into another asset
<b>annotate</b>	The act of adding notations/commentaries to the asset creating a new asset
<b>aggregate</b>	using an asset (or parts of it) as part of a composite work or collection.

Table 5.4: ODRL permissions (all definitions start with *the act of...*)



three principal sections, although the highest, root, level also has attributes which enables the specification of the kind of rights being described, e.g. copyrighted, licensed, public domain, contractual, or other. These are the main sections:

- RightsDeclaration, a broad declaration of the rights associated with a digital asset intended to inform the user community of these rights.
- RightsHolder, details of any person or organization holding some rights to a given digital asset. Contains subelements to specify rightsholder details: the name of the rights holder, the name of the person or organization acting as a contact for the rightsholder, along with contact addresses, telephone numbers and email addresses.
- Context, describes the specific circumstances associated with who has what permissions and constraints, matching largely what was named as conditions in the other RELs.

With the METSRights schema, specific permissions can be stated (see Table 5.5), but on the contrary to MPEG-21 rights or ODRL permissions, they are not XML elements but attributes. This disallows making extensions. The rights can be given only in a certain context, e.g. quality, format, payment, re-use, other. While it is a well founded language, it supplies no novelty respect the other RELs and no use has been found out of the libraries in the USA. Actually, no automated control over use is intended with METSRights, and the data elements do not support such control. The audience for the METSRights elements is an information professional rather than a machine.

### TV Anytime

TV-Anytime is a set of specifications (initiated by the *TV-Anytime Forum*) for the controlled delivery of multimedia content to a user's digital video recorder. The specification was closed in 2005, and soon after sanctioned by the ETSI (European Telecommunications Standards Institute).

TV-Anytime is the metadata standard reference for video broadcasting, but it has also an insight for DRM systems and a Working Group on Rights Management and Protection (RMP) was established to enable the secure and flexible expression and enforcement of rights holders' usage, achieving a specification published in May 2009 [124].

<i>Right</i>	<i>Definition</i>
discover	Resource is available for searching or other discovery
display	Rendering, playing, executing the resource
copy	Making verbatim copy for purposes of re-use of whole or part of the resource and creation of new resource
duplicate	Make exact copy of resource for file or repository management purposes
modify	Annotate, edit, excerpt, embed, extract resource for purposes of re-use or preservation
delete	Remove resource from repository for purposes of resource or repository management
print	Rendering the resource onto paper or hard copy
other	

Table 5.5: METS rights

In it, TV-Anytime speaks about RMPI (Rights Management and Protection Information) or, *the minimum set of usage rules and conditions required to enable protection of broadcast digital television content within a TVA RMP compliant domain*, and it defines four elements to represent the conditions and constraints to access multimedia content. The elements defined are *grants*, *principals*, *rights* and *conditions*, words already known in the MPEG-21 REL vocabulary.

*Principal* distinguish between the *receiving domain* and *any domain*. The former is the first TVA RMP-compliant domain that receives the content and the later is any TVA RMP-compliant domain that can respond to the usage conditions stated within RMPI-MB and RMPI-M.

*Rights* are those listed in Table 5.6.

*Conditions* follow here:

- Geographical Control: This condition limits the use of a right to within one or more specified territories.
- Single Point of Control: This condition limits the use of a right to a particular domain.
- Physical Proximity: This condition limits the use of a right to RMP compliant devices within close physical proximity of the receiver that first received the broadcast content.
- Buffer Duration: This condition limits the use of a right in such a way that each frame of broadcast content is used only within a specified duration after that frame was broadcast.

<i>Right</i>	<i>Definition</i>
Play	right to derive a transient and directly perceivable representation of content within the TVA RMP domain
Analogue Export	right to create a user accessible analogue signal representing the content as an output, and side of TVA RMP system
Digital Exp. Standard	right to create a Standard Definition digital signal representing the content as an output outside of the TVA RMP system
Digital Exp. High Def.	right to create a High Definition digital signal representing the content as an output outside of the TVA RMP system
Extend Rights	right for the RMP System to apply additional rights to the content. The absence of this right means that only the originally transmitted rights may be applied

Table 5.6: TV-Anytime rights

- Time Window Start Date and Time Window End Date: These conditions define the window of time during which the rights are granted. It is defined as absolute start time and absolute expiry time.
- Standard Definition Digital Export Control: This condition forwards content management rules to external content protection systems on standard definition outputs. The content can be viewed as well as recorded or stored.
- High Definition Digital Export Control: This condition forwards content management rules to external content protection systems on high definition outputs whilst exercising the Digital Export HD right: for immediate viewing only bound to device or media for future viewing.
- Analogue Export Signaling: This condition forwards content management rules to external content protection systems: for immediate viewing only bound to device or media for future viewing (includes immediate viewing).
- Analogue Standard Definition (SD) control: This condition constrains the resolution of the exported analogue signal.
- Security Level: This condition constrains the execution of rights based on the invoked components' robustness level. Security levels are to be based upon the aggregate robustness of all invoked components needed to exercise a right.

- Simultaneous Rendering Count: This condition limits the number of simultaneous Plays, Analogue Exports and Digital Exports of content within a domain.
- Source of additional rights: This condition identifies the authority which may assign new rights to the content.

Of special interest for this work is the TV-Anytime value chain in Section 3 of [124], which declares as participants to the following actors: Content Creator, Rights Owner, Content Provider, Advertisers, Distributors, Service Providers, Portal Owners, Network Operators, Equipment Manufacturer and retailer, Consumer, and Metadata Providers.

### **Creative Commons licenses**

Creative Commons (CC) licenses are by far the most famous amongst the RELs described in this Section, almost every experience internaut knows what a creative commons license is. In December 2002, a non-profit organization called Creative Commons released its first set of copyright licenses for free to the public; being among the members of the founding board Lawrence Lessig (author of [1], for example). The licenses, inspired largely by the GPL philosophy succeeded soon and spread very fast because of their simplicity.

A simple logo attached to the protected resource could be dropped in a web page and have some warranties of protection. There was no intention to prevent the resource to be used or copied, as DRM systems make with other RELs, but there was the intention of stating precisely how could this be done without risk of legal prosecution. This was a great thing for small companies and individual creators, which could not pay a swarm of lawyers but could easily add a logo in their PDF, photo or whatever the resource was. The success was so high that soon after non-web resources also incorporated the CC license. Nevertheless, it has to be remarked that CC licenses collide with certain legal systems, for example the Spanish: authors cannot give up their exploitation rights.

These are the possible symbols that define the Creative Commons license:

- Attribution (by): Licensees may copy, distribute, display and perform the work and make derivative works based on it only if they give the author or licensor the credits in the manner specified by these.

- Noncommercial or NonCommercial (nc): Licensees may copy, distribute, display, and perform the work and make derivative works based on it only for noncommercial purposes.
- No Derivative Works or NoDerivs (nd): Licensees may copy, distribute, display and perform only verbatim copies of the work, not derivative works based on it.
- ShareAlike (sa): Licensees may distribute derivative works only under a license identical to the license that governs the original work.

The alleged objective of the project was to promote creativity by setting a large collection of CC licensed material available in the web usable for new creations for free, and is for this regard that all creative commons licenses permitted using the work for free. Undoubtedly, the objectives have been largely accomplished.

Bizarre as it might appear, a simple drawing accompanying a song changes completely its legal status, and this has been recognized in the courts. For the Spanish case, this has happened repeated times.

The paradox is that for the purposes of this Thesis, the Creative Commons licenses are not directly interesting because they are not computer readable. They are actually in a binary form, but in their simplest -and most widespread- form they are the few bytes of an image that computer applications cannot read and understand (rather than drawing it). These licenses are to address persons and not computers. In fact, the Creative Commons graphic or logo that is attached to the resource or published in the web page serves as a mere reference to the actual narrative license found on the CC web page.

However, one of the projects started later on within the Creative Commons organization was the ccREL (Creative Commons REL). This was the response to the demand of having a computer readable REL equivalent to the CC licenses. Creative Commons Rights Expression Language (ccREL) is specified in [125], and it describes how license information may be expressed using RDF and how license information can be attached to works (as an annexed RDF file, or embedded in the XMP of a PDF etc.). Creative Commons plans creating registries to keep an account of the CC protected material.

### Coloriuris

A similar and more recent initiative in Spain is *Coloriuris*<sup>12</sup>, with the difference that there is a company behind it. Initially addressed to bloggers, it is not free, there is a registry with a timestamp attached at registration time which adds increased chances in case of conflict -conflict where the user is defended by the Coloriuris lawyers. In opposition to Creative Commons, whose terms collide frontally with the continental legal traditions (that is to say, those more influenced by the *droit d'auteur* than the copyright), Coloriuris acknowledges the moral rights of the author which are never dropped (as it happens in CC).

Being the business model thus different, the REL they use -and what is of interest for this Thesis, is not specially original: CC symbols are substituted by colours, whose combination can determine up to 11 different contracts, with different combinations of options for the economical exploitation, the derived works, the public communication etc.

### Specialized RELs

MPEG-21 REL or ODRL are examples of rights expression languages with a general scope, and part of their complexity is that they have to cover every imaginable right and constraint. There are many other RELs, which have been developed ad-hoc for particular needs. For example, the PDF documents readers developed by Adobe are able to read and interpret a set of rights codified in the document, in a non standard REL expression. Some PDF documents can be read but not printed, or some forms in PDF format can be filled-in but not stored, and even some can not be even seen without a proper license. These rights (print, store, read) and some others are not codified using a normalized expression but a private one and they are not subject of study in this work.

#### 5.3.4 Semantic RELs

There have been other attempts at least from the academia to build RELs not at a syntactic XML but at a semantic level. Although they have not gained acceptance, they are in the line of this work.

The most important references are LicenseScript [126], which is based in Logic Programming (see Section 3.1.2) and with a materialization in Prolog.

---

<sup>12</sup>Coloriuris, autogestión de derechos de autor y registro de obras, <http://www.-coloriuris.net/>

OREL [127] was a well founded proposal for a REL in OWL, but it did not introduce any element essentially different from those in MPEG-21 REL; and it remained a mere academic proposal. A direct mapping of MPEG-21 REL and MPEG-21 RDD were the ontologies RDDOnto and RELOnto introduced in [128].

OntologyX, a formal ontology for media rights transactions was an initiative from the industry (Rightscom) but whose public description [129] has been limited, in contrast with the aforementioned for which there is a public specification file.





**Part II**  
**Contribution**



## Chapter 6

# The Media Value Chain Ontology

This Chapter describes the Media Value Chain Ontology, the spinal cord of this Thesis. The Media Value Chain Ontology is the representation in Description Logics of the workflow of multimedia content and the successive transfers of intellectual property. This work has been acknowledged by the standardization bodies and currently is in the process of becoming an ISO/IEC standard.

### 6.1 Industrial Value Chain and Intellectual Value Chain

In our modern, highly organised society, goods are the result of complex processes rather than the production of a single man. For each of our industrial products, it can be traced a long succession of steps adding value to the good, and transforming it until it is consumed in a final form. This chain of economical transactions is acknowledged and taxed by governments (e.g. VAT, value added tax).

The amount of people needed to work for one to play a compact disc is countless, if it is considered the cover, the lyrics and the disc itself and how complex is the production of each of them, including chemical industries for the colour ink of the workart, woodstocks for the paper and so on. In practice, the industrial chain is untraceable, or rather, it fades in participations of infinitesimal value.

However, the artistic creations are not automated tasks able to be reproduced in series<sup>1</sup> -the quintessential requisite for industrial production- and producing art as a craftsman's work necessarily implies few participants. The artistic creation conveyed in the disc mentioned above is the result of a few minds -perhaps a composer, an arranger, an interpreter and some sound engineers. It is true that each intellectual creation is a social product, in the sense that each work has been inspired or influenced from the society achievements (e.g. the musical artists that had influenced the composer), without which that music would have never existed, but the contribution of an artist can also be well defined and separated. Thus, the intellectual value chain can be traced with very few and precise steps.

## 6.2 The Intellectual Property Value Chain Model

The intellectual property value chain refers to how value is added to the IP objects through the links by certain actors performing certain actions. The value chain can be seen as a chain of objects, a chain of people or a chain of actions. These basic elements (objects, users and actions) have a fundamental relation shown in Fig. 6.1. A User (subject) acts Action (verb), and Action is acted over an IP entity (object). Finally, the IP entity has a User as rights owner.

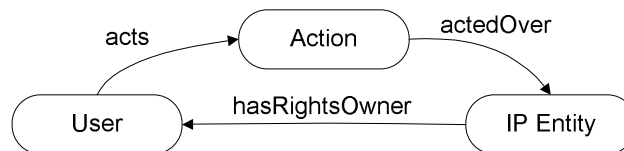


Figure 6.1: Relations between actions, users and IP entities

It is acknowledged in the treaties and laws that the origin of the intellectual property value chain is in the creator of a work. The act of creation is the starting point of the chain. Of course, no creation is absolutely original and every work is almost completely determined consciously or not by the cultural substrate of the author or authors. But precisely it is the author's last differential contribution that is most appreciated and the only that can be accurately attributed. Thus, the vision of this Thesis agrees in establishing the work as the origin of the value chain and his author as the first agent.

<sup>1</sup>Visions like Noll's [130], with the computer as the center in the creation process, can be ignored for the moment.

The work and the successively derived objects in the value chain are called *IP Entities*. IP entities are all the different kind of objects that can be subject to Intellectual Property. This categorization is independent of the media type (e.g. audio, video, text), and even the work is only of an abstract nature.

Thus, the pristine result of the creation process is called the *work*, and it is the first object of the value chain. The work is an abstract idea, residing originally only in the author's mind, and which to be communicated to others has to be materialized in a physical representation (a *manifestation*). The creator of the idea is sometimes referred as *author*, and either individually or collectively it represents a unit.

If the work is not original and it has been derived substantially from another work, it is referred as *adaptation*. Adaptations are actually works whose provenance is another work. Adaptation authors require of course permission from the original creator to make the derived entity.

Creators (either of original *works* or *adaptations*) express their ideas through *manifestations* of their work, the first material representation of the IP object. *Manifestations* can be music scores, drafts, descriptions, tentative interpretations or any other descriptive form of the work. Note that while *works* or *adaptations* are mere conceptual entities, manifestations are physical objects.

Note that this concept of manifestation is different from the concept of manifestation in FRBR (Section 4.2.1). It cannot be assimilated either to the FRBR concept of *expression*, being the concept of 'FRBR expression' wider than 'intellectual property manifestation'. A theater piece interpretation would be a FRBR expression, but not a 'intellectual property manifestation', which would be the libretto (musical script).

Music compositions, theater pieces and other IP entities are susceptible of being played, interpreted or performed many times yielding *instances*. The person who carries it out will be called *instantiator*. An *instance* is the permanent representation of an execution, and the result of fixating an execution of a work. The *first fixation* of a work has a special legal treatment and it is a synonym of instance. It will be the material that a producer will take for making *copies* arranging them properly. Finally, collection of copies making *products* will be distributed to an *end user* in order to enjoy it.

A manifestation in FRBR, defined as 'physical embodiment of one or more expressions', matches the concept of 'intellectual property copy', and the FRBR 'item' the concept of 'product'.

Figure 6.2 shows the three categories of elements conforming the most relevant steps in the intellectual property value chain. Some persons (*roles*)

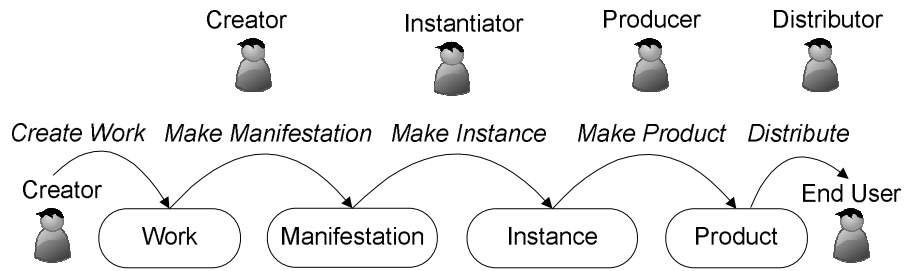


Figure 6.2: Main stages in the IP value chain

can be seen, as well as some verbs (that will be called *actions*) and some types of IP objects (*IP entities*) in boxes.

The author has all the rights over his work. These rights can be passed to another person, due to the author's death (*mortis causa*) or due to a voluntary decision (*inter vivos*). Some moral rights can not be waived, and cannot be transferred, but the rest can be transferred at author's will. This transfer may include all the rights or only some of them, and can be in exclusive to the licensee or not. Exclusive transfer usually allows the licensee to relicense the rights to a third person.

Note that other emerging value chains, significantly abbreviated, may look to lack some of the intermediate steps, but it is a false impression. In the case that a composition has an interpretation by the author which is directly distributed through online services like social networks (Fig. 6.3), an each time a more frequent case) the author plays the roles of creator, instantiator and producer; and he has the set of all the corresponding rights for each of the roles.

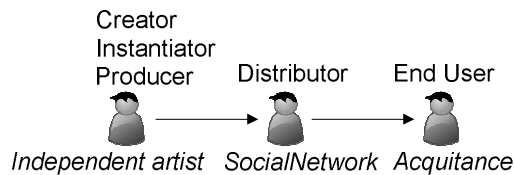


Figure 6.3: Shortened value chain and accumulation of roles

### 6.2.1 IP entities

IP entities are one of the three basic categories of entities the model deals with (Fig. 6.1). The enumeration of the different IP entities which can be

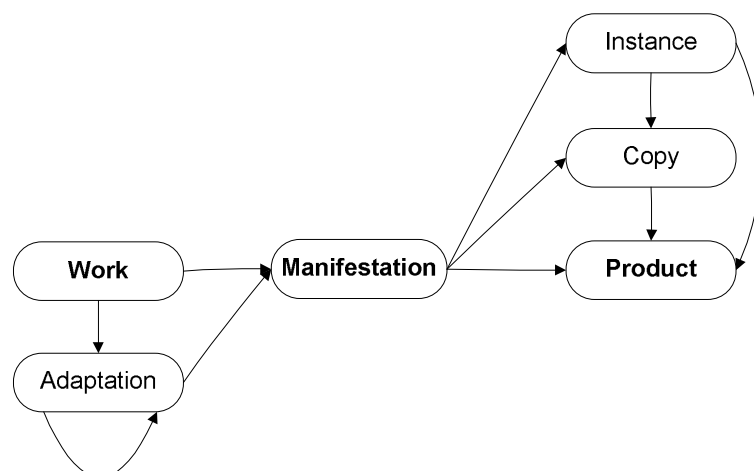


Figure 6.4: IP Entities transitions in the Value Chain

recognized in the transformation process from work to product are listed here:

**Work** An original abstract idea that can be uniquely attributable.

**Adaptation** A work that is based on another work.

**Manifestation** The tangible physical expression of a work such as a musical score, manuscript or event that can be recorded.

**Instance** A particular execution or rendition of a manifestation.

**Copy** A copy of an instance or a manifestation, equal to other copies.

**Product** A collection of one or more copies ready to be distributed.

The legal transitions from one IP entity to other entity are shown in Fig. 6.4. A work necessarily has to go through the step of manifestation (it has to be somehow materially expressed). Note that works can be adapted or not. Before reaching the end user, the manifestation, which is a unique material item, has to be multiplied and adopt the form of product (if the edition is of 1000 books, there are 1000 products). Optionally, there are possible steps in the middle: an interpretation different from that of the author (a different performer artist appears), arrangements in format and form (sound engineers, cover artists etc.) etc.

Each of the listed IP entities can be further refined. For example, we can observe that there are different manifestations, depending on if they come from a work or from an adaptation etc. We can name then *WorkManifestation* and *AdaptationManifestation* respectively etc. Each of these concepts will be further detailed in the next sections.

### 6.2.2 Users

Users are the agents which interact with the IP entities. They are not necessarily persons, they can also be institutions or machines, and they can play different roles at the same time. The individuals that act on these basic IP entities can be classified according to a set of generic roles that can be adopted by an agent i.e. a person or group thereof who incarnates one or more roles. User roles are the second big category of entities this model deals with. The list of roles is the following:

**Creator.**<sup>2</sup> The author of the work, who translates his idea into a material realization.

**Adaptor.**<sup>3</sup> The creator of an adaptation from a work.

**Instantiator.**<sup>4</sup> An agent who executes a performance or rendition of the work.

**Producer.** An agent who compiles commercial distributable products.

**Distributor.**<sup>5</sup> An agent who distributes the product.

**End User.** The last agent to use the content.

This set of core roles can be extended to add further specializations not included in the Value Chain model. For example, in the music context, it

---

<sup>2</sup>The definition found in the Spanish Ley de Propiedad Intelectual (LPI) is: *Se considera autor a la persona natural que crea alguna obra literaria, artística o científica.*

<sup>3</sup>The definition found in the Spanish LPI is given by the person who makes: *Las traducciones y adaptaciones. Las revisiones, actualizaciones y anotaciones. Los compendios, resúmenes y extractos. Los arreglos musicales. Cualesquiera transformaciones de una obra literaria, artística o científica.*

<sup>4</sup>The definition found in the Spanish LPI is: *Se entiende por artista intérprete o ejecutante a la persona que represente, cante, lea, recite, interprete o ejecute en cualquier forma una obra.*

<sup>5</sup>In the Spanish LPI, a distributor is the person who makes distribution, and *Se entiende por distribución la puesta a disposición del público del original o copias de la obra mediante su venta, alquiler, préstamo o de cualquier otra forma.*



would make sense deriving a new concept called *Composer*, derived from Creator and inheriting all its features, but also having new characteristics specific to music composers.

The basic *leitmotif* of this Thesis is that the author of an idea is the owner of the rights over it, and that this rights can be traded. Each time the IP Object passes from one person to another, more added value is added, and it is the object of our study the general (and economical) terms by which transfers of right take place.

### 6.2.3 Actions

In our context, actions are the processes of doing something over IP entities that is relevant to the Intellectual Property. The actions defined in the Value Chain model are:

- Action to create an original IP entity: Create a work.
- Actions to create dependent IP entities: Make an adaptation, a manifestation, an instance, a copy or a product. It is also included the synchronization of two distinct instances each for a different sense (e.g. and audio and text, video and soundtrack, etc.).
- Actions to communicate IP entities: Make a public communication, a live performance, broadcasting or any other form of distribution.
- Actions to consume IP entities: Play, print or any other rendering action.

Each action can be exercised over only one kind of IP entity, and it can only be performed by one User role. It has to be remarked, that this categorization of actions has only been made explicit here, as the conceptual model had never made such a distinction. One kind of action is of particular interest: the actions that generate new IP entities, which are depicted in Fig. 6.5.

The complete value chain with IP entities, user roles and actions is seen in Fig. 6.6.

### 6.2.4 Permissions

The conceptual IP model also presents about how rights are traded. Rights may be transferred with exclusivity or not, and some may be resold or not. The creator may retain rights, and the execution of certain actions

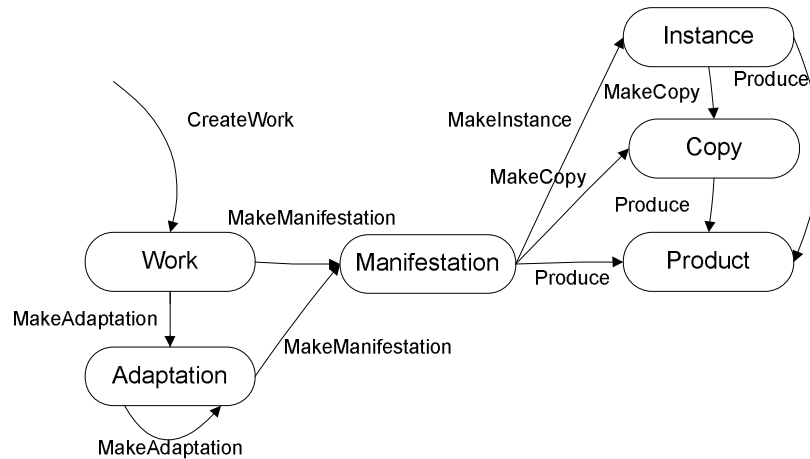


Figure 6.5: The IP Entities in the Value Chain and their originating actions

require his approval through transfer of the corresponding rights. In many cases not all of the roles intervene, and the requirements for the transfer of rights may differ, but these differences should be capable of being expressed as particular specialization of the model implemented as extensions. In occasions, rights that the creator cannot waive include the right to perceive royalties (an income proportional to the number of items sold to the end user) and in general all the moral rights.

The User who performs an action has to hold the right for executing it over a given IP entity, and this ownership is transferred by virtue of Permissions. Note that whereas exploitation rights can be transferred, author's moral right cannot, and authors will have to additionally give the consent for certain operations at the end of the Value Chain, like the public communication of a work instance.

A *permission* relates an IP entity with the transferred right, the original rights owner and the new rights owner (see Fig. 6.7), being only valid if the given requirements are satisfied.

These permissions complement the licenses defined making use of MPEG-21 Part 5, the Rights Expression Language (REL). REL licenses represent a set of permissions that can be expressed in many ways in the context of extensions of the MVCO.

In the next section, the representation of this model will be precised and developed into its details.

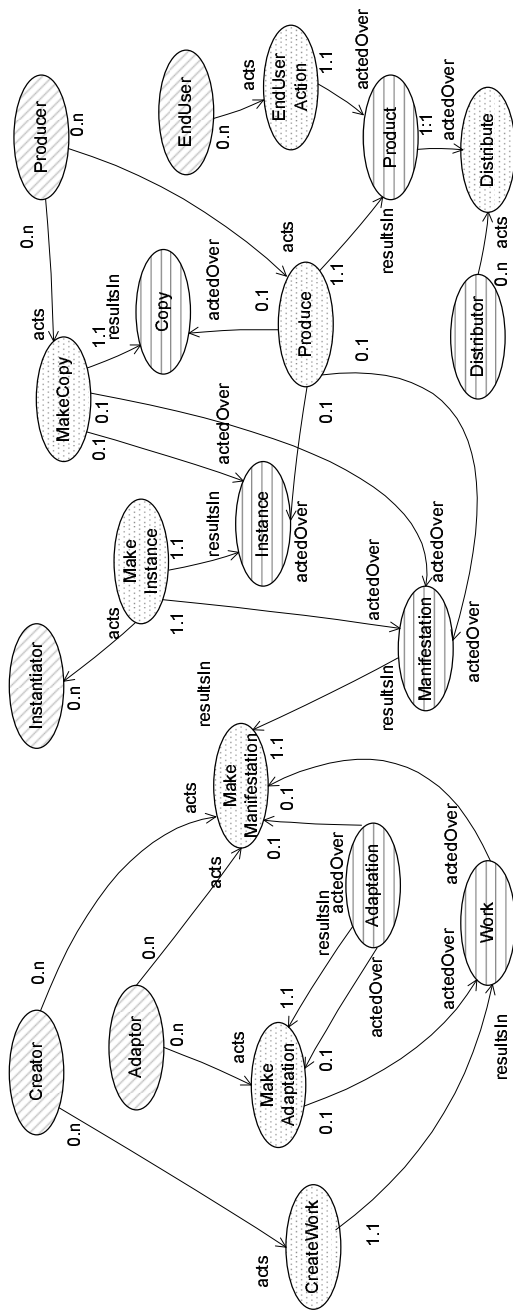


Figure 6.6: Complete value chain of IP entities, user roles and actions

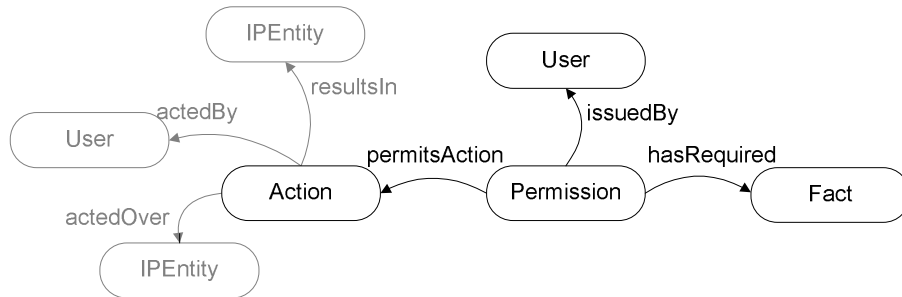


Figure 6.7: Permissions in the Value Chain

### 6.3 The Media Value Chain Ontology

The model specified in the previous section can be codified with the elements of Section 3.3 as an OWL ontology: this is the Media Value Chain Ontology. This section presents the Media Value Chain Ontology with OWL excerpts in Turtle syntax [39] commented in English, along with a description of the development process, the paradigmatic scenario and several examples of use.

#### 6.3.1 Ontology development

The representation as explicit knowledge of the implicit facts conveyed in the model, those extracted from contacts and those needed to perform some basic operations, is not an immediate task but rather an iterative procedure where the codified gets refined progressively. This task of developing an ontology has been studied and some good methodologies collections have been described ([131], [132], [133]). This Thesis has followed the ideas proposed in [134], which lists a number of steps to be followed:

1. Determine the domain and scope of the ontology. This step has been described in the Introduction (Section 1.1).
2. Consider reusing existing ontologies. Reusing ontologies in practice means either borrowing terms from similar domain ontologies or basing new concepts on upper ontologies. Both practices have been avoided for this work in order to keep simplicity at maximum and minimize the dependencies.
3. Enumerate important terms in the ontology. Terminology was largely given by legal texts and existing dictionaries, nevertheless, key terms

had to be chosen and some of them reformulated. They have been informally listed in Section 6.2.

4. Define the classes and the class hierarchy. This is the concern of this Section.
5. Define the properties of classes. This is the concern of this Section too.
6. Define the additional properties related to or necessary for properties (i.e., cardinality, bidirectionality/inverse, etc.).
7. Create instances. This ontology has defined only one instance, a User individual named `mvco:anonymous` and used to refer to the anonymous agent.
8. Create axioms/rules.

The MVCO ontology is the only accurate representation of the value chain, and this Section is merely the text description of the same. The rest of the section describes the MVCO.

### 6.3.2 Namespaces and metadata

The ontology defines its own namespace as a permanent URL (PURL). Permanent URLs can redirect dynamically to different locations while keeping the same name, and they are commonly used in other ontologies like Dublin-Core. The MVCO URI is:

```
http://purl.oclc.org/NET/mvco.owl#
```

Elements with no prefix are supposed to belong to MVCO's own namespace (see lines 1-2 in Listing 6.1). Lines 3-9 show the namespaces needed in the ontology to reference the OWL language constructs. Lines 10-11 include namespaces defined in the MPEG standards. Dublin Core annotations (ISO 15836-2003) have been made to the ontology, and the elements `dc:title` and `dc:language` have been used, therefore the DublinCore namespace is also needed (line 12).

Listing 6.1: Namespace declarations in the MVCO

```
1 @prefix : <http://purl.oclc.org/NET/mvco.owl#> .  
2 @prefix :mvco <http://purl.oclc.org/NET/mvco.owl#> .  
3 @prefix :xsd <http://www.w3.org/2001/XMLSchema#> .
```

```

4 | @prefix :rdf <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
5 | @prefix :rdfs <http://www.w3.org/2000/01/rdf-schema#> .
6 | @prefix :owl <http://www.w3.org/2002/07/owl#> .
7 | @prefix :swrl <http://www.w3.org/2003/11/swrl#> .
8 | @prefix :swrlb <http://www.w3.org/2003/11/swrlb#> .
9 | @prefix :mx <urn:mpeg:mpeg21:2003:01-REL-MX-NS#> .
10| @prefix :dii <urn:mpeg:mpeg21:2002:01-DII-NS#> .
11| @prefix :dc <http://purl.org/dc/elements/1.1/> .

```

### 6.3.3 Ontology classes

#### Class list

This section lists the classes defined as `owl:Class`.

Textual definitions are given as `rdfs:comment` comments, in the English language (`xml:lang="en"`). The subclassing information is also given here (made with the construct `rdfs:subClassOf`). The author of this Thesis should not be attributed the merit of precisising the English definitions which follow. That was an arduous process in which the author was only one voice more. But the existence itself of the elements and its hierarchical arrangement is a direct result of his work.

Each of the classes was given a version number.

```
owl:versionInfo "1.0"^^http://www.w3.org/2001/XMLSchema#string;
```

The classes in Listing 6.2 are the `mvco:User` class and its derived subclasses.

Listing 6.2: User class and subclasses

```

:User a owl:Class;
  rdfs:comment "Any person or legal entity in a Value-Chain
  connecting (and including) Creator and EndUser."@en.
:Collective a owl:Class;
  rdfs:comment "Set of two or more Users."@en;
  rdfs:subClassOf :User .
:Creator a owl:Class;
  rdfs:comment "A User who generates a Work and makes its
  first Manifestation, also referred to as author"@en;
  rdfs:subClassOf :User .
:Adaptor a owl:Class;
  rdfs:comment "A User who produces an Adaptation"@en;
  rdfs:subClassOf :User .
:Instantiator a owl:Class;
  rdfs:comment "A User who interprets a Manifestation of a
  Work making an Instance"@en;

```

```

    rdfs:subClassOf :User .
:Producer a owl:Class;
    rdfs:comment "A User who produces a Product from an
Instance."@en;
    rdfs:subClassOf :User .
:Distributor a owl:Class;
    rdfs:comment "A User who distributes a Product"@en;
    rdfs:subClassOf :User .
:EndUser a owl:Class;
    rdfs:comment "A User in a Value-Chain who ultimately
consumes Content."@en;
    rdfs:subClassOf :User .

```

Listing 6.3 show the mvco:IPEntity class and its derived subclasses.

Listing 6.3: IP Entity class and its subclasses

```

:IPEntity a owl:Class;
    rdfs:comment "Types of IP Represented as Content: Work,
Adaptation, Manifestation, Instance..."@en;
:Work a owl:Class;
    rdfs:comment "A creation that retains intellectual or ar-
tistic attributes independently of its Manifestations"@en;
    rdfs:subClassOf :IPEntity .
:Adaptation a owl:Class;
    rdfs:comment "A Work that is derived from another Work"@en;
    rdfs:subClassOf :IPEntity .
:Manifestation a owl:Class;
    rdfs:comment "An object or event which is an expression of
a Work."@en;
    rdfs:subClassOf :IPEntity .
:Instance a owl:Class;
    rdfs:comment "An object or event which is an example of an
Identified Manifestation (e.g. a File)"@en;
:Copy a owl:Class;
    rdfs:comment "A mechanical reproduction of analogue or
digital representations of a given IP Entity. In the case of
digital Copies the result is virtually identical while in the
case of analogue Copies the results can vary considerably in
quality."@en;
    rdfs:subClassOf :IPEntity .
:Product a owl:Class;
    rdfs:comment "A Content Item that adds value to IP Entities
by including them with an appropriate Licence for the purpose
of Publishing"@en;
    rdfs:subClassOf :IPEntity .
:UseData a owl:Class;
    rdfs:comment "Data documenting the functions Actions
performed by a Device User on a content item and the
associated context"@en;

```

```

    rdfs:subClassOf :IPEntity .
:AdaptationManifestation a owl:Class;
    rdfs:comment "An object or event which is an expression of
an Adaptation"@en;
    rdfs:subClassOf :Manifestation .
:WorkManifestation a owl:Class;
    rdfs:comment "An object or event which is an expression of
a Manifestation of a Work"@en;
    rdfs:subClassOf :Manifestation .
:WorkInstance a owl:Class;
    rdfs:comment "An object or event which is an example of an
Identified Manifestation of a Work (e.g. a File)"@en;
    rdfs:subClassOf :Instance .
:AdaptationInstance a owl:Class;
    rdfs:comment "An object or event which is an example of an
Identified Adaptation Manifestation (e.g. a File)"@en;
    rdfs:subClassOf :Instance .
:AdaptationInstanceCopy a owl:Class;
    rdfs:comment "A copy of an AdaptationInstance"@en;
    rdfs:subClassOf :Copy .
:AdaptationManifestationCopy a owl:Class;
    rdfs:comment "A copy of an AdaptationManifestation"@en;
    rdfs:subClassOf :Copy .
:WorkInstanceCopy a owl:Class;
    rdfs:comment "A copy of a WorkInstance"@en;
    rdfs:subClassOf :Copy .
:WorkManifestationCopy a owl:Class;
    rdfs:comment "A copy of a WorkManifestation."@en;
    rdfs:subClassOf :Copy .

```

Listing 6.4 show the mvco:Action class and its derived subclasses.

Listing 6.4: Action class and its subclasses

```

:Action a owl:Class;
    rdfs:comment "The exercise of a Right."@en;
:CreateWork a owl:Class;
    rdfs:comment "The Action of creating a Work."@en;
    rdfs:subClassOf :Action .
:MakeAdaptation a owl:Class;
    rdfs:comment "The Action of making an Adaptation"@en;
    rdfs:subClassOf :Action .
:MakeManifestation a owl:Class;
    rdfs:comment "The Action of making a Manifestation
."@en;
    rdfs:subClassOf :Action .
:MakeInstance a owl:Class;
    rdfs:comment "The Action of making an Instance from a
Manifestation."@en;
    rdfs:subClassOf :Action .

```



```

:MakeCopy a owl:Class;
  rdfs:comment "The Action of making a Copy"@en;
  rdfs:subClassOf :Action .
:Produce a owl:Class;
  rdfs:comment "The Function of making Products"@en;
  rdfs:subClassOf :Action .
:Synchronise a owl:Class;
  rdfs:comment "The Action of concurrently performing/
  displaying two distinct IP Entities each for a different
  human sense e.g. text and audio or video and song"@en;
:PublicCommunication a owl:Class;
  rdfs:comment "The Action of publicly displaying/performing
  , e.g. live performance, radio, television, internet
  streaming, multicast of Instances and Manifestations,
  and download"@en;
  rdfs:subClassOf :Action .
:Distribute a owl:Class;
  rdfs:comment "The Action of selling, renting and lending
  "@en;
  rdfs:subClassOf :Action .
:EndUserAction a owl:Class;
  rdfs:comment "Action performed by an End User"@en;
  rdfs:subClassOf :Action .
:Broadcast a owl:Class;
  rdfs:comment "The Function that Delivers Content to
  a Device in a point-to-multipoint modality"@en;
  rdfs:subClassOf :PublicCommunication .
:Download a owl:Class;
  rdfs:comment "The Action of transferring a file or program
  from a central computer to a smaller computer or to a
  computer at a remote location"@en;
  rdfs:subClassOf :PublicCommunication .
:Stream a owl:Class;
  rdfs:comment "The Action of Delivering Content to a Device
  where the transferred Content is Processed for Rendering only
  and not Stored"@en;
  rdfs:subClassOf :PublicCommunication .
:MakeAdaptationInstanceCopy a owl:Class;
  rdfs:comment "The Action of making an AdaptationInstance-
  Copy"@en;
  rdfs:subClassOf :MakeCopy .
:MakeWorkInstanceCopy a owl:Class;
  rdfs:comment "The Action of making an WorkInstanceCopy"
  @en;
  rdfs:subClassOf :MakeCopy .
:MakeWorkManifestationCopy a owl:Class;
  rdfs:comment "The Action of making an WorkManifestation-
  Copy"@en;
  rdfs:subClassOf :MakeCopy .

```

```

:MakeAdaptationManifestationCopy a owl:Class;
  rdfs:comment "The Action of making an AdaptationManifestationCopy"@en;
  rdfs:subClassOf :MakeCopy .
:MakeAdaptationInstance a owl:Class;
  rdfs:comment "The Action of making an Instance from an AdaptationManifestation"@en;
  rdfs:subClassOf :MakeInstance .
:MakeWorkInstance a owl:Class;
  rdfs:comment "The Action of making an Instance from a WorkManifestation"@en;
  rdfs:subClassOf :MakeInstance .
:MakeWorkManifestation a owl:Class;
  rdfs:comment "The action of making a Manifestation from a Work."@en;
  rdfs:subClassOf :MakeManifestation .
:MakeAdaptationManifestation a owl:Class;
  rdfs:comment "The action of making a Manifestation from a Manifestation Work."@en;
  rdfs:subClassOf :MakeManifestation .
:Fact a owl:Class;
  rdfs:comment "Positive proposition."@en;
:Permission a owl:Class;
  rdfs:comment "Authorisation from one RightsOwner to one or more Users to perform one or more Actions on a given IPEntity."@en;

```

### Equivalence class assertions

Some of the presented classes are *defined classes* and not *primitive classes*. In OWL, the construct `owl:equivalentClass` is used, which in Turtle notation is simply represented with the symbol ‘=’ (Listing 6.5).

Listing 6.5: Defined classes in the MVCO

```

:Creator = [:acts owl:someValuesFrom :CreateWork] .
:Adaptor = [:acts owl:someValuesFrom :MakeAdaptation] .
:Instantiator = [:acts owl:someValuesFrom :MakeInstance] .
:Producer = [:acts owl:someValuesFrom :Produce] .
:Distributor = [:acts owl:someValuesFrom :Distribute] .
:EndUser = [:acts owl:someValuesFrom :EndUserAction] .
:Fact = [:isTrue owl:cardinality 1] .

```

Each of these expressions could have been interpreted as:

$$\text{Creator} \equiv \exists \text{ acts CreateWork}$$

etc.

### 6.3.4 Ontology properties

#### Object properties

This section enumerates the properties defined as `owl:ObjectProperty`. Their definition is again given as `rdfs:comments`. Each object property has been given a range and a domain, expressed with the `rdfs:domain` and `rdfs:range` constructs.

Listing 6.6: Functional object properties in MVCO

```

:hasRightsOwner rdf:type owl:FunctionalProperty ,
                  owl:ObjectProperty ;
                  rdfs:comment "Defines the owner of the Rights
over an IP Entity."@en ;
                  rdfs:domain :IPEntity ;
                  rdfs:range :User .
:resultsIn rdf:type owl:FunctionalProperty ,
             owl:ObjectProperty ;
             rdfs:comment "Declares which IPentity arises as a
result of the execution of an Action."@en ;
             rdfs:domain :Action ;
             rdfs:range :IPEntity .

```

Listing 6.6 enumerates the functional object properties, i.e. those which can be attributed only once to one individual. This is represented with the OWL construct `owl:FunctionalProperty`. This is key for modelling the `mvco:hasRightsOwner` property: IP entities can only have one owner.

Listing 6.7 enumerates the non-functional object properties. Two of them are transitive, like the belonging of a member to a collective relationship (`owl:TransitiveProperty`). Additionally, it can be read that the object property `mvco:hasRightsOwner` is the inverse of `mvco:isRightsOwnerOf` and `mvco:resultedFrom` is the inverse of `mvco:resultedIn` (`owl:inverseOf`).

Listing 6.7: Non-functional object properties in MVCO

```

:acts rdf:type owl:ObjectProperty ;
       rdfs:comment "Performance of an Action by a User"@en ;
       rdfs:range :Action ;
       rdfs:domain :User ;
       owl:inverseOf :actedBy .
:actedBy rdf:type owl:ObjectProperty ;
          rdfs:comment "Role who has executed the Action"@en ;
          rdfs:domain :Action ;
          rdfs:range :User .
:actedOver rdf:type owl:ObjectProperty ;
            rdfs:comment "Specifies which IPentity is the object

```

```

        the Action"@en ;
        rdfs:domain :Action ;
        rdfs:range :IPEntity .
:belongsTo rdf:type owl:ObjectProperty ,
            owl:TransitiveProperty ;
        rdfs:comment "Relates a User with a Collective"@en ;
        rdfs:range :Collective ;
        rdfs:domain :User .
:hasRequired rdf:type owl:ObjectProperty ;
        rdfs:comment "For a Permission to be valid, the
        Fact has to hold"@en ;
        rdfs:range :Fact ;
        rdfs:domain :Permission .
:isMadeUpOf rdf:type owl:ObjectProperty ;
        rdfs:comment "Relates a composite IPEntity with its
        constituent IPEntities"@en ;
        rdfs:domain :IPEntity ;
        rdfs:range :IPEntity .
:isRightsOwnerOf rdf:type owl:ObjectProperty ;
        rdfs:comment "Declares who is the rights owner
        of this IP Entity"@en ;
        rdfs:range :IPEntity ;
        rdfs:domain :User ;
        owl:inverseOf :hasRightsOwner .
:issuedBy rdf:type owl:ObjectProperty ;
        rdfs:comment "Declares who has issued a
        permission"@en
        rdfs:domain :Permission ;
        rdfs:range :User .
:permitsAction rdf:type owl:ObjectProperty ;
        rdfs:comment "Relation used to express the Ac-
        tions that are allowed to be performed."@en ;
        rdfs:range :Action ;
        rdfs:domain :Permission .
:resultedFrom rdf:type owl:ObjectProperty ;
        rdfs:comment "The dependance of one IP Entity on
        another."@en ;
        rdfs:range :Action ;
        rdfs:domain :IPEntity ;
        owl:inverseOf :resultsIn .
:actOnBehalfOf rdf:type owl:ObjectProperty ,
            owl:TransitiveProperty ;
        rdfs:comment "Relates a ContentHandler with the
        User under the auspices of which the
        ContentHandler operates."@en ;
        rdfs:domain :ContentHandler ;
        rdfs:range :User .

```

### Datatype properties

Listing 6.8 enumerates the properties defined as `owl:DatatypeProperty`, useful to attribute data values to OWL class individuals.

The first one, `dii:RelatedIdentifier`, enables the direct linking of IP Entities to MPEG Digital Items. As it has been defined in the ontology, one IP Entity can only be related to one Digital Item. The DublinCore element `dc:title` is used to title the ontology. `isTrue` is the key boolean attribute associated to a `mvco:Fact` class.

Notably, the datatype property `mvco:hasSocialTag` in the MVCO was proposed by ETRI, intended to give social attributes to either the IP Entities or the MVCO Users. This has been patented [135].

Listing 6.8: Datatype object properties in MVCO

```

DII-NS:RelatedIdentifier rdf:type owl:DatatypeProperty ,
                          owl:FunctionalProperty ;
                          rdfs:comment "It allows the identification
information that is related to a
Digital Item (or parts thereof)."@en ;
                          rdfs:domain :IPEntity ;
                          rdfs:range xsd:string .

:hasSocialTag rdf:type owl:DatatypeProperty ;
rdfs:comment "Attributes a social tag: text,
commentary, critique, synopsis (e.g.
FOAF)..."@en ;
rdfs:range xsd:string ;
rdfs:domain [ rdf:type owl:Class ;
              owl:unionOf ( :IPEntity
                             :User
                           )
            ] .

:isDigital rdf:type owl:DatatypeProperty ,
            owl:FunctionalProperty ;
            rdfs:comment "distinguishes between digital ma-
nagement of both digital and non digital IP En-
tities"@en ;
            rdfs:domain :IPEntity ;
            rdfs:range xsd:boolean .

:isTrue rdf:type owl:DatatypeProperty ,
          owl:FunctionalProperty ;
          owl:versionInfo "1.0"@en ;
          rdfs:comment "Truth of a proposition"@en ;
          rdfs:domain :Fact ;
          rdfs:range xsd:boolean .

dc:title rdf:type owl:DatatypeProperty ,
          owl:FunctionalProperty ;
          rdfs:comment "Title of the ontology"@en ;

```

```
rdfs:range xsd:string .
```

### Annotation properties

Annotation properties can be given to OWL elements, but they will not be used for reasoning, nor will their consistency be checked. Object properties and datatype properties can only be attributed to OWL individuals if the OWL language has to remain at a DL level. However, annotation properties can be attributed to classes or other properties without surpassing the DL level: precisely because they are formally disregarded for reasoning.

Two annotation properties have been defined in the MVCO ontology (Listing 6.9): the first is `impliesAlso`, used to express that having the rights to execute one Action implies also having the rights to execute the second (e.g. to make a Synchronise it is necessary to make a Render). The second, `rightsGivenBy`, declares who is the user role that has to authorise a certain action.

Listing 6.9: Datatype object properties in MVCO

```
:impliesAlso rdf:type owl:AnnotationProperty .
:rightsGivenBy rdf:type owl:AnnotationProperty .
```

### 6.3.5 Ontology restrictions

This subsection describes the restrictions which apply over the different classes. Restrictions add strength to the model, stating clearly what is against the model and what ontology extensions are not allowed. Restrictions in Turtle syntax appear like the one given in Listing 6.10, however, for the sake of clarity, the rest of the subsection will use the formal logic syntax.

Listing 6.10: Restrictions over IPEntity in Turtle syntax

```
:IPEntity rdfs:subClassOf [ a owl:Restriction;
  owl:onProperty
  :hasRightsOwner; owl:allValuesFrom :User
  owl:cardinality 1
  :resultedFrom; owl:allValuesFrom :Action
  :isMadeUpOf; owl:allValuesFrom :IPEntity];
```

An IP entity has one and only one User as rights owner, that a composite IP entity may be made of several IP entities and that IP entities are created as results of actions:

<p> <math>\text{IPEntity} \subseteq \forall \text{ hasRightsOwner } \text{User}</math>  <math>\text{IPEntity} \subseteq \text{hasRightsOwner exactly } 1</math>  <math>\text{IPEntity} \subseteq \text{isMadeUpOf } \text{IPEntity}</math>  <math>\text{IPEntity} \subseteq \text{resultedFrom } \text{Action}</math> </p>
--

IPEntities derive from the act that created them. A Work is only the result of a CreateWork Action. The other IPEntities must have a precedence, thus a Manifestation comes from at least MakeManifestation, an Adaptation can only derive either from the execution of a MakeAdaptation or a Synchronise action etc.

<p> <math>\text{Work} \subseteq \forall \text{ resultedFrom } \text{CreateWork}</math>  <math>\text{Adaptation} \subseteq \exists \text{ resultedFrom } \text{MakeAdaptation} \cup \text{Synchronise}</math>  <math>\text{Manifestation} \subseteq \exists \text{ resultedFrom } \text{MakeManifestation}</math>  <math>\text{Instance} \subseteq \exists \text{ resultedFrom } \text{MakeInstance}</math>  <math>\text{Copy} \subseteq \exists \text{ resultedFrom } \text{MakeCopy}</math>  <math>\text{Product} \subseteq \exists \text{ resultedFrom } \text{Produce}</math> </p>
---

Users may only act Actions. This prevents the ontology to be extended in malicious ways where other verbs can be acted rather than Actions.

<p> <math>\text{User} \subseteq \forall \text{ acts } \text{Action}.</math> </p>
--

Actions are acted by one and only one User. They may imply other actions (for example a public communication implies making a rendering). Additionally and to exclude other classes from the control mechanisms, only users can execute actions, and only users can give rights to execute actions.

<p> <math>\text{Action} \subseteq \exists \text{ actedBy } \text{User}</math>  <math>\text{Action} \subseteq \text{actedBy exactly } 1</math>  <math>\text{Action} \subseteq \exists \text{ impliesAlso } \text{Action}</math>  <math>\text{Action} \subseteq \exists \text{ rightsGivenBy } \text{User}</math> </p>
--

The action of creation of a Work is the first action over an IPEntity. This has been stressed in the model, and thus in the ontology it is explicitly specified that no one can give the right to create a Work, that CreateWork cannot be executed over any IPEntity and that the result of creating a work is one Work (and at most one, because resultsIn is a functional relation).

<p> <math>\text{CreateWork} \subseteq \neg ( \exists \text{ actedOver } \text{IPEntity} )</math>  <math>\text{CreateWork} \subseteq \text{resultsIn } \text{Work}</math>  <math>\text{CreateWork} \subseteq \text{rightGivenBy exactly } 0</math> </p>
--

An Adaptation has to be made over a Work. Only Creators give the right to make an Adaptation. The act of making an adaptation results in one Adaptation.

$\text{MakeAdaptation} \sqsubseteq \exists ( \text{ actedOver } \text{Work} )$ $\text{MakeAdaptation} \sqsubseteq \exists ( \text{ rightGivenBy } \text{Creator} )$ $\text{MakeAdaptation} \sqsubseteq \exists ( \text{ resultsIn } \text{Adaptation} )$
--

A Manifestation has to be done over some Work or Adaptation. The act of making a manifestation results in one Manifestation.

$\text{MakeManifestation} \sqsubseteq \exists ( \text{ actedOver } ( \text{Work} \cup \text{Adaptation} ) )$ $\text{MakeManifestation} \sqsubseteq \exists ( \text{ resultsIn } \text{Manifestation} )$
--

Instances are created from Manifestations. The act of making an Instance results in one Instance.

$\text{MakeInstance} \sqsubseteq \exists ( \text{ actedOver } \text{Manifestation} )$ $\text{MakeInstance} \sqsubseteq \exists ( \text{ resultsIn } \text{Instance} )$
---

Copies are created from Instances. The act of making a Copy results in Copy.

$\text{MakeCopy} \sqsubseteq \exists ( \text{ actedOver } \text{Instance} )$ $\text{MakeCopy} \sqsubseteq \exists ( \text{ resultsIn } \text{Copy} )$
--

Products are created from Copies, Instances or even Manifestations. Producing requires the additional permission of the Creator, what is a discordant feature over the chain of permissions but reflects the legality and the international agreements.

$\text{Produce} \sqsubseteq \exists ( \text{ actedOver } ( \text{Copy} \cup \text{Instance} \cup \text{Manifestation} ) )$ $\text{Produce} \sqsubseteq \exists ( \text{ resultsIn } \text{Product} )$ $\text{Produce} \sqsubseteq \exists ( \text{ rightsGivenBy } \text{Creator} )$
--

Synchronization is made over Works. Making a Synchronization requires the additional permission given by the Creator, and any User who has the right to Synchronise, can also invoke the right to Render or Modify a Copy.

$\text{Synchrhonise} \sqsubseteq \exists ( \text{ actedOver } \text{Work} )$ $\text{Synchrhonise} \sqsubseteq \exists ( \text{ rightGivenBy } \text{Creator} )$ $\text{Synchrhonise} \sqsubseteq \forall \text{ impliesAlso } ( \text{Render} \cup \text{ModifyCopy} )$
---

Public Communication can be performed on Copies or Products. Making PublicCommunication requires the additional permission given by the Creator. Any User who has the right to make a PublicCommunication, can indeed invoke the right to Render.

$\text{PublicCommunication} \sqsubseteq \exists ( \text{ actedOver } \text{Copy} \cup \text{Product} )$ $\text{PublicCommunication} \sqsubseteq \exists ( \text{ rightGivenBy } \text{Creator} )$ $\text{PublicCommunication} \sqsubseteq \forall \text{ impliesAlso } \text{Render} )$
---



There must be a Product to distribute. The act of Distribute cannot generate any additional IPEntity and distributing requires the additional permission of the Creator.

Distribute  $\subseteq \exists$  ( actedOver Product )  
 Distribute  $\subseteq$  ( resultsIn exactly 0 )  
 Distribute  $\subseteq \exists$  ( rightsGivenBy ( Creator  $\cup$  Producer ) )

It is defined that EndUsers only deal with Products. Also, EndUserActions cannot generate any additional IPEntity

EndUserAction  $\subseteq \exists$  ( actedOver Product )  
 EndUserAction  $\subseteq$  resultsIn exactly 0

Permissions permit at least one Action and only Actions. Permissions are issued by exactly one User. Permissions have as requirements only Facts.

Permission  $\subseteq \exists$  permitsAction Action  
 Permission  $\subseteq \forall$  permitsAction Action  
 Permission  $\subseteq \exists$  issuedBy User  
 Permission  $\subseteq$  issuedBy exactly 1  
 Permission  $\subseteq \exists$  hasRequired Fact

Also as a basic feature of the model some restrictions `owl:disjointWith` on the disjointness of classes have been stated. For example, Action, User, IPEntity and Permission are mutually disjoint: an Action can never be an IPEntity, and a Permission is not an User etc.

### Ontology features

MVCO expressivity is  $\mathcal{SIF}(\mathcal{D})$ , where S represents a transitive Base Description Logic (Attributive Language with Complements (transitive ALC)),  $\mathcal{I}$  represents that inverse operators are used in the ontology,  $\mathcal{F}$  represents the functional attribute of relations (that is to say, when a relation can at most be only once present for an individual) and ( $\mathcal{D}$ ) represents the use of datatypes (see Section 3.2.1).

## 6.4 MVCO evolution

The MVCO ontology emerged in its first form with the name *Rights Represents Data Ontology* or RRD as an input proposal for its adoption in the Digital Media Project as a part of the Interoperable DRM Platform (IDP) (see entry #1 in Table 6.1), and after a few months of informal work. This

version drafted the current Media Value Chain Ontology, and included some elements that later would disappear: the private copy was represented, as well as the mechanical reproductions. Also, roles were pre-assigned to the users instead of leaving them freedom to play any role.

The requirements (entry #4 in Table 6.1) and the ontology itself were formally adopted by the DMP in its 13th General Assembly (#5) resulting in changes in the Approved Documents N°2, N°3 and N°7 (whose latest versions are [102] [103] [107]). A tentative API had also been developed on top of the ontology so that Java applications making use of the ontology could be fast prototyped, as well as an accompanying demonstration software (entries #2 and #3). A software showing the ontology capabilities was officially held in the 14th General Assembly (#7).

Small refinements in the ontology would include the REL rights (entry #6 in Table 6.1) and cardinality in the relations. Synchronization was reshaped allowing to be performed on two IP Entities. These changes were ready for the next version of the prior to Interoperable Platform, IDP v3.0 in the General Assembly 15 (#8). The final version of RRDOnto is labeled as 6.4 and it can be found in the latest IDP specification, the version 3.2 [107].

The ontology at this stage was presented in [136], and a glimpse of the ontology at this stage can be seen in Fig. 6.8, created with the tool GrOWL [137].

In parallel, the Media Value Chain Ontology had been proposed in the context of a larger project for the cross media distribution, the AX4HOME (to be described in Section 8.2). The set of requirements were defined in May 2007 (entry #9 in Table 6.1) and the ontology itself, named AxIPOntology, formally incorporated to the project soon after (entry #10). For the first time, it had been integrated with the rest of the elements in a DRM architecture. The API was implemented, and a server application was developed on top of it (this will be described in Section 8.2). The latest version of the ontology (AxIPOntology 4.0) was delivered in September 2007, and a demonstration of the software capabilities was done one month later (entry #11). Compared to the RRD, AxIPOntology was a bit simpler, more oriented towards a practical implementation where some unnecessary restrictions were removed (like the `owl:disjointWith`).

The API also included more functions. A very similar version of this ontology was adapted and used in the context of the E2E project (entry #13 in Table 6.1, and in detail in 8.3).

Later on, a call for requirements was presented in the MPEG group to develop a Media Value Chain [152] [153] [154]. The next version of the work was presented to the call with the name IPMNet (entry #14 in Table 6.1).

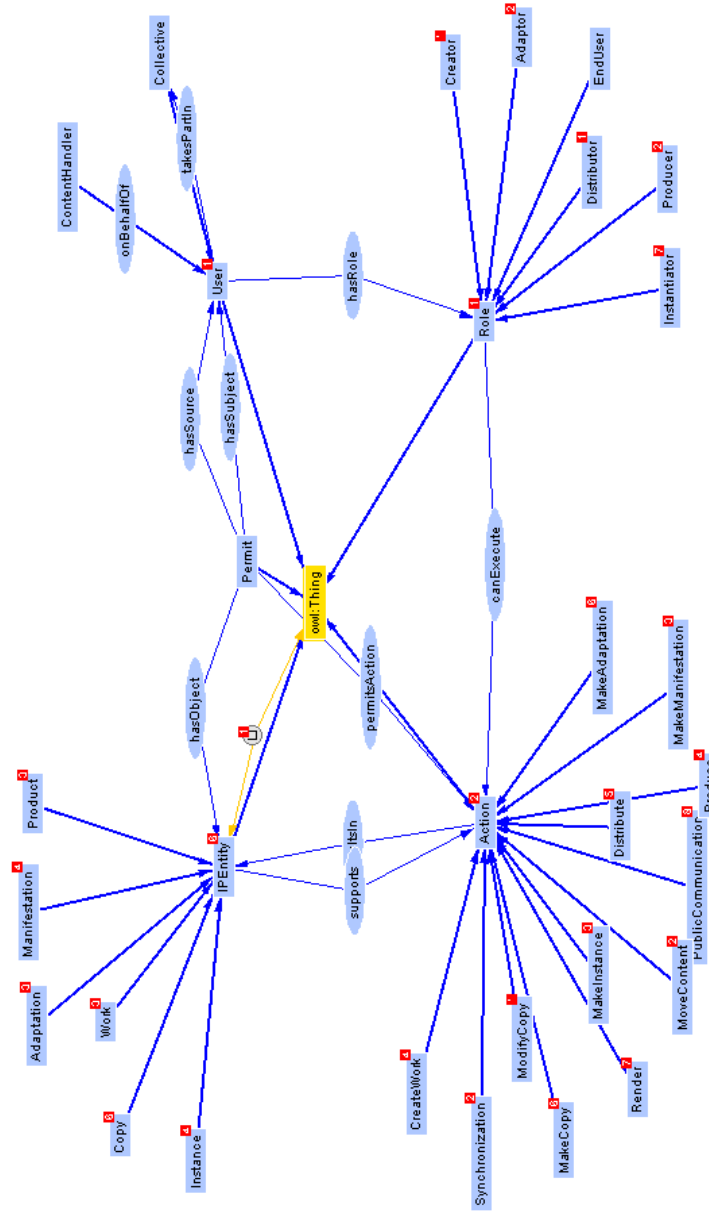


Figure 6.8: Early Media Value Chain Ontology

#	Date	Event	Ref.
1	04/10/2006	1st public version of the ontology	[138]
2	02/11/2006	1st version of the API	
3	11/01/2007	1st demo software	
4	26/01/2007	Requirements for RRD	[139]
5	02/02/2007	Ontology as a part of IDP 2.1	[140]
6	07/03/2007	Refinements in the ontology for GA14	[141]
7	15/05/2007	Demo held in GA14 meeting	
8	20/07/2007	Ontology as a part of IDP 3.0	[142]
9	07/05/2007	Ontology proposed in Axmedis requirements	[143]
10	13/07/2007	AxIPOntology in Axmedis architecture	[144]
11	25/10/2007	Demo of AxIPOntology	
12	25/04/2008	The ontology in the E2E architecture	[145]
13	12/09/2008	E2EOnto delivered in E2E	
14	09/07/2008	Ontology responding to MPEG call as IPMNet	[146]
15	25/07/2008	MVCO WD1.0	[147]
16	20/10/2008	MVCO CD1.0	[148]
17	07/11/2008	MVCO Introduction Web page	web <sup>6</sup>
18	06/02/2009	MVCO FCD1.0	[149]
19	19/07/2009	MVCO FCD2.0	[150]
20	29/01/2010	MVCO FDIS1.0	[151]

Table 6.1: MVCO milestones and evolution

IPMNet had renamed most of the relations and many of their classes. Also the definition of the classes had been changed; if RRD had definitions given by the DMP, IPMNet had definitions by their own, more akin to the MPEG language.

Soon after the ontology was approved to satisfy the requirements, with some minor additions from another proposal ([155]), it was accepted and named Media Value Chain Ontology (entry #15), labeled successively 0.1 in the first Working Draft stage of the standardization process, 0.2 in the Committee Draft stage of the standardization process (entry #16) and 0.3 in the Final Committee Draft stage, which comprised two editions (entry #18 and #19). Between all these version, only minor corrections were made, and some definitions were changed after bitter controversy. Between them, a divulgative web had been open (#17) with an interesting mapping to the Music Ontology (Section 4.2.1). As of February 2010, the First Draft of International Standard has been written (#20), the ontology is labelled as 1.0, and it is waiting for balloting. The ontology presented in this Chapter is the last one, being the most evolved and perfect from all the previous. The text in the standard, however, was redacted amid tense debates and an

environment of pressure where changes could not be carried out as readily as would have been desired. The author of this Thesis believes that the text in this Dissertation is superior in clarity and rigour to that in the ISO/IEC standards.

### Ontology versioning

Classes and properties have been annotated with the `owl:versionInfo` tag. This tag version should be 1.0 in the first edition of the International Standard, and possible corrigenda to the standard should upgrade the decimal (e.g. 1.1, 1.2 etc.), while possible amendments would increase the version number (e.g. 2.0 etc.). Further amendments of this standard may introduce new classes and relationships with higher versions, but they should not deprecate the existing axioms (e.g. no `owl:class` will be turned out into `owl:deprecatedclass` etc.).

## 6.5 Alignment and extension mechanisms

The Semantic Web allows any entity to post, reuse and extend OWL ontologies to suit their needs. MVCO foresees extensions of the ontology, for which the following extension mechanisms are listed:

- Addition of new classes. Subclasses can be created and inherit all the features of the parent class while allowing refining concepts for more specialised purposes.
- Adding new relations. New object properties can be defined, either at root level or derived from other object properties.
- Adding new properties. New datatype properties can be defined at will.
- Creating individuals. The extended ontology may define individuals.

Any ontology extending MVCO that is consistent can be said to be MVCO compatible. The built-in MVCO properties and classes should not be redefined. In general, this means that elements from the `mvco` namespace should not appear as subjects of triples (only as objects).

### Ontology alignment

Other ontologies may want to map some of their concepts to those in MVCO. In order to make a conceptual mapping, SKOS should be used (Section 3.3.2), but in order to make a working mapping able to safeguard the logic tractability of the representation, OWL constructs should be used -and only some OWL constructs. For example, individuals of one ontology may be mapped to individuals of another, through the use of the construct `owl:-sameAs`, but classes should not be matched given that if classes were matched with `owl:sameAs` the ontology would result in a OWL Full and would become intractable.

Ontology alignment is useful in scenarios sharing some concepts with the MVCO and applications may use both ontologies having matched certain terms. For example, the Music Ontology focuses in a particular media - Music - and defines a workflow whose main concepts can be assimilated by some concepts in the MVCO.

The next exemplifies an MVCO extension and it describes how the CreativeCommons model can be represented with the MVCO ontology. The base has been taken from a RDF representation itself of CreativeCommons licenses, given by CC itself<sup>7</sup>.

#### 6.5.1 CreativeCommons: An example of MVCO extension

The RDF version of CreativeCommons describes a set of classes and properties, and for each of them, the mapping is given here.

##### CreativeCommons classes

Some of the CreativeCommon classes have a direct equivalence with the MVCO classes. If the extension was to make reasoning only inheritance or comments could be used; if not, `owl:sameAs` should suffice.

1. `cc:Work` a potentially copyrightable work. Same as in `mvco:IPEntity`
2. `cc:License` a set of requests/permissions to users of a Work, e.g. a copyright license, the public domain, information for distributors Same as `mvco:Permit`
3. `cc:Jurisdiction` the legal jurisdiction of a license. A new datatype `mvco:hasCountry` has been created. This attribute can be given to attribute Permissions to individuals.

<sup>7</sup>Describing Copyright in RDF [online] <http://creativecommons.org/ns>

4. **cc:Permission** an action that may or may not be allowed or desired It is an Action taking part in a Permit. It is a Defined Class . Class name: **cc:Permission**. Necessary: **subClassOf Action Necessary** and sufficient:  $\exists$  **isPermittedIn . Permit**

4.1 **cc:Reproduction** making multiple copies. Same as **mvco:MakeCopy**.

4.2 **cc:Distribution** distribution, public display, and publicly performance Either same as **mvco:Distribute** or **mvco:PublicRepresentation**.

4.3 **cc:Derivative Works** distribution of derivative works. Same as **mvco:MakeAdaptation**.

4.4 **cc:HighIncomeNationUse** use in a non-developing country. MVCO extension **mvco:isHighIncomeNation** datatype boolean property has been created in. Permits can include conditions based on this attribute.

4.5 **cc:Sharing** permits the use of the derivatives, but only for non-commercial distribution. It is the same as **mvco:Distribute** but for a zero price.

5. **cc:Requirement** an action that may or may not be requested of you. It is a Fact taking part in a Permit as requirement (through the relation **mvco:isRequiredIn**). It is a defined class. Class name: **cc:Permission**. Necessary: **subClassOf Fact Necessary** and sufficient:  $\exists$  **isRequiredIn . Permission**

5.1 Notice copyright and license notices be kept intact A subclass of Fact called **mvco:CopyrightNotice** has been created. When true, the given copyright notice is present and the **mvco:Permission** can be validated.

5.2 Attribution credit be given to copyright holder and/or author A subclass of Fact called **mvco:Attribution** has been created. When true, the attribution message is present and the **mvco:Permission** can be validated.

5.3 Share Alike derivative works be licensed under the same terms as the original work. A subclass of Fact called **mvco:ShareAlike** has been created. A SWRL rule could check that all the derived works have this attribute present.

5.4 Source Code source code (the preferred form for making modifications) must be provided for all derivative works A subclass of Fact called **mvco:SourceCode** has been created. A SWRL rule could check that all the derived works have this attribute present.

6. **cc:Prohibition** something you may be asked not to do It is a Fact taking part in a **mvco:Permission** as prohibition (through the relation **mvco:isProhibitedIn**). It is a defined class. Class name: **cc:Prohibition** . Necessary: **subClassOf Fact Necessary** and sufficient:  $\exists$  **isProhibitedIn . Permit**

6.1 Commercial Use exercising rights for commercial purposes A subclass of `mvco:Fact` called `mvco:CommercialUse` has been created.

### CreativeCommons properties

#### Work Properties

1. `cc:morePermissions` The URL where additional permissions or alternative licenses for a Work are available. Mapped to a datatype property
2. `cc:attributionName` The name the creator of a Work would like used when attributing re-use Mapped to a datatype property
3. `cc:attributionURL` The URL the creator of a Work would like used when attributing re-use. Mapped to a datatype property
4. `cc:jurisdiction` A License may have a jurisdiction, as defined by Jurisdictions. Mapped to a datatype property
5. `cc:legalcode` The URL of the legal text of a License. Mapped to a datatype property
6. `cc:deprecatedOn` A License may be deprecated; provides the date deprecated on. Mapped to a datatype property

#### LicenseProperties

1. `cc:licenseA` Work has license a License. (a subproperty of `dc:license`, the same as `xhtml:license`)

The Permits are related to the Actions through the object property `mvco:permitsAction`. In turn, Actions are related to IP Entities with the object property `mvco:executedOver`. The automated mapping cannot be expressed with OWL but with SWRL or through SPARQL queries.

2. `cc:permits` A License permits a Permission. Same as the MVCO property `mvco:permitsAction`.
3. `cc:requires` A License requires a Requirement. Same as the MVCO property `mvco:hasRequirement`.
4. `cc:prohibits` A License prohibits a Prohibition Same as the property `mvco:hasProhibited`.

## 6.6 Scenario of use

### 6.6.1 The MVCO in a Content Distribution System

By making explicit the knowledge of the IP model along the value chain, the MVCO satisfies a double purpose: it lays down a precise reference to solve disputes between the different actors, and it serves as a basis to develop interoperable applications. The sample scenario of use in an application –but



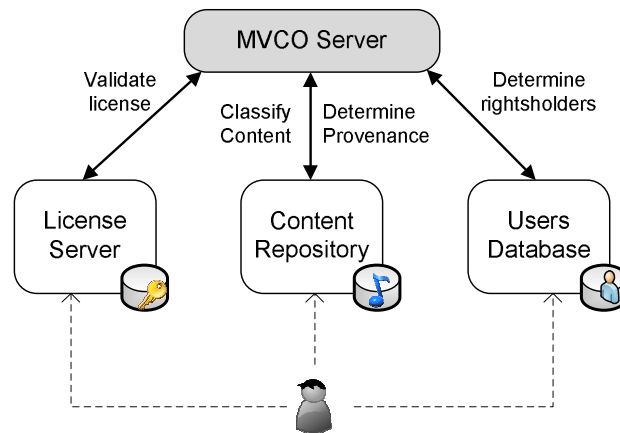


Figure 6.9: MVCO Server in a simplistic distribution architecture

not the only one— places the MVCO at the heart of a content distribution system in which the value chain users interoperate with content repositories, license providers and user authentication systems (see Fig. 6.9).

In this simple scenario, some users inject new IP protected content as Digital Items, and some users (the rights holders) issue licenses for other users to use the content or become in turn new rights holders. The MPEG-21 standard provides the framework to deploy these architectures, and the MVCO Part 19 comes to fill a gap and complement the existing technologies by facilitating the following functionalities:

- Attribute the kind of IP that a Digital Item represents (e.g. whether a certain resource represents a manifestation or an instance). Also, to link an IP entity to the preceding IP entities in the value chain (e.g. from which work an adaptation comes from). More than an additional column in the content repository database, the expression of this information in the MVCO grants that the IP schema has been respected, as logical verifications can be made to grant the overall consistency of the IP model.
- Define the role that a user plays regarding a given IP entity. This information is changing along the time (e.g. the creator of a work may become also the performer of an instance of the work etc.) according to the actions that are being done.
- Keep track of who is the legal rights owner of each IP entity at any moment, given that this status changes through permissions and agree-

ments. These permissions can be expressed also in the MVCO following the IP model (e.g. to authorise broadcasting certain content, the permission of both the work creator and instantiator are needed). Verify that the MPEG-21 REL licenses that are granted are respectful to the value chain (i.e. the license issuer match with the rights owner etc.)

Each of these three listed functionalities exceeds the natural scope of the three elements (content repository, license server, and users registry) in Fig. 6.9, and the MVCO provides their proper integration in the matters related to the IP.

### 6.6.2 Using MVCO: a walkthrough

This section assumes the scenario given in the previous section, trying to facilitate understanding how to practically exploit the MVCO.

Consider the the Content Management System given before with the following components:

- Content Repository, possibly storing MPEG-21 Digital Items.
- A Licensing Server, possibly storing and authorising MPEG-21 REL licenses.
- A User database, including content providers, instantiators, broadcasters, end users etc.
- A MVCO Server

Consider Bob an identified User Creator of an original Work with a digital Manifestation encapsulated in a DI. The process is as follows:

1. The Work is registered in the MVCO Server, being Bob the sole Rights owner.
2. The Manifestation is registered in the MVCO Server, being linked with the Work it is of.
3. The DI embodying the Manifestation is registered and uploaded in the Content Repository.
4. A REL license template is uploaded into the Licensing Server

Now consider a second User of the system, called Alice, who is an Instantiator and wants to make an instance of Bob's work.

1. Alice obtains the DI representing Bob's Work Manifestation with a REL license only for creating an Instance (an appropriate Right can be defined).
2. Alice uploads the Instance embodied in a Digital Item.

Finally we consider a third User of the system, `BroadcasterUnion`, a broadcaster who is interested in making a Public Communication of Alice's performance of Bob's Work. Note that in order to make a Public Communication, both the consent of the performer and that of the author are required.

1. `BroadcasterUnion` obtains the DI, by buying a REL license
2. `BroadcasterUnion` obtains a Permission from the MVCO Server, issued by Alice as the Instantiator
3. `BroadcasterUnion` obtains a Permission from the MVCO Server, issued by Bob as the Creator.

The MVCO Server can be used for a number of purposes:

- Certify that licenses and license templates adhere to the Value Chain Model. In particular:
  - Certify that Bob's REL license template is valid, as it is issued for a Manifestation whose Work belongs to him.
  - Certify that Alice's Instances belongs to her.
- Validate operations whose authorisation mechanism is out of MPEG-21 REL scope. In particular:
  - Validate that `BroadcasterUnion`'s broadcasting counts with the permissions of both interpreter and author and thus is conformant to the IP Value Chain.
- Respond to queries in some of the richest expression formats, like SPARQL, taking advantage of MVCO Server reasoning capabilities. Thanks to MVCO inferencing potential, intelligent responses can be given. For example:

Answer queries about which kind of IP Entity a Digital Item represents.

Answer queries about which kind of Role a User plays with respect to a certain IP Entity.

Answer queries about relationships between IP Entities like provenance, existing permissions and derivative works.

- Offer the best interfaces for interoperability and ontology extension. For example:

Offer an interface of the system to other platforms in the easiest possible fashion. Semantic Web expressions are thought and intended for interoperation with other Semantic Web models, thanks to its semantic expression capabilities. Mappings are expressed as RDF / OWL and seamlessly integrated in the system.

Grant compatibility between custom MVCO extensions refining existing classes or properties.

Note that these functions cannot be performed by virtue of any of the existing parts of MPEG-21 or combinations thereof.

### 6.6.3 Class individuals in several MVCO use cases

For several cases, this section describes which class individuals are present at a given moment in the ontology.

#### Bob creates a Work

OWL individuals in the MVCO represent real Users, IP Entities, Actions and Permissions. When Bob creates a Work, the ontology represents it with a minimum of two MVCO class individuals: Bob and the created Work. But the Action itself can also be represented and stored as an individual. This Action may trigger an Event Report (MPEG-21 Part 15).

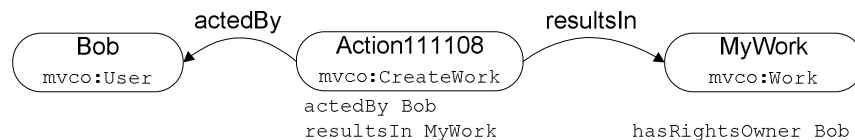


Figure 6.10: Three class individuals in the ontology

Fig. 6.10 represents the three individuals, “Bob” ,“Action111108” and “MyWork” and the class they belong (User, CreateWork, Work). The arrows represents the relations (object properties) by which they are linked, and below the values taken by the object properties are given. Note that not all of the possible object properties are represented here.

### Alice makes an Instance

Once Bob has created a Work and a Manifestation (the latter not represented here), Alice’s Action of creating an Instance is represented in Fig. 6.11. Note that in MVCO some properties have exactly one value. Thus, the object property ‘actedBy’ has to receive a value for each existing Action, while ‘actedOver’ may or may not. Also note that Fig 6.11 represents the ‘Action11110’ as an individual of MakeInstance.

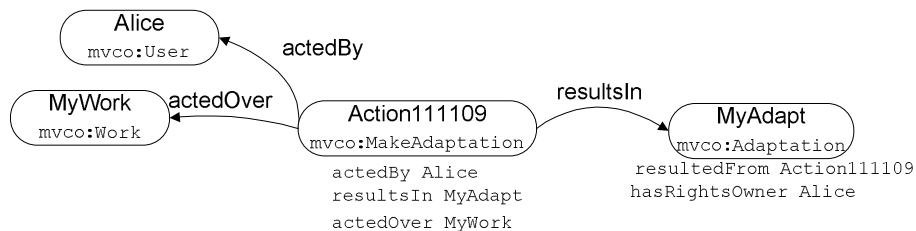


Figure 6.11: Individuals when Alice makes an Instance

### Charles and Claire make a PublicCommunication

Finally, Charles and Claire, acting as a Collective (called BroadcasterUnion) make a PublicCommunication of Bob’s Work, instantiated by Alice (Fig. 6.12).

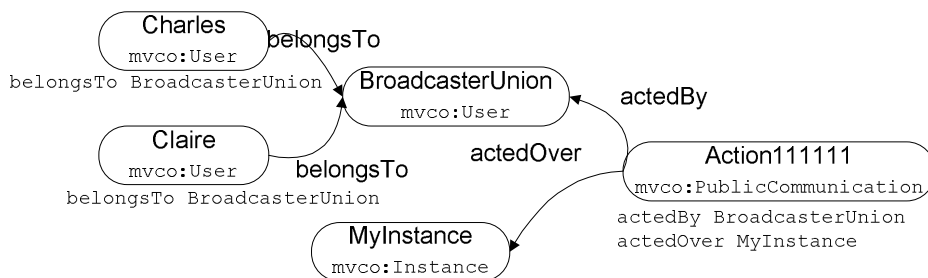


Figure 6.12: Alice’s Instance is broadcasted

### Charles and Clair have Permission to make the PublicCommunication

For the Public Communication of Bob’s Work to adhere to the Value Chain Model, there must be two Permissions; not only the one from Alice as Interpreter (Fig. 6.14) but also from Bob as the Creator (Fig. 6.13).

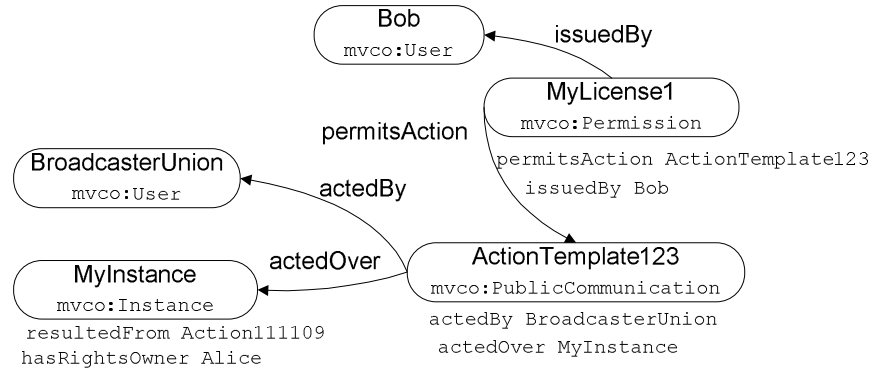


Figure 6.13: Bob's authorisation

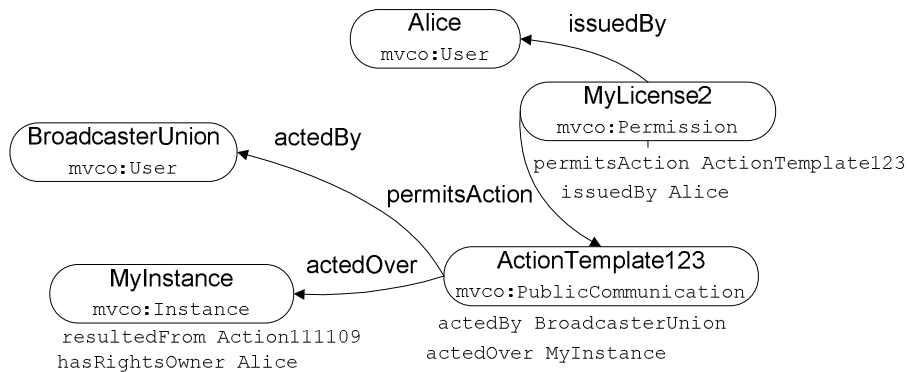


Figure 6.14: Alice's authorisation

## 6.7 Example of a rights transfer with MVCO

This Section gives an example showing the MVCO permission equivalent to a MPEG-21 REL license. First, it will be done using only the the elements in the core MVCO. Then, a second example will show how an MVCO extension can be used to represent all the elements in the REL license.

### 6.7.1 An exemplar license

The sample license considered here is the first one appearing in the REL standard (in [97] Annex D.2, *Simple end-user License example*), where Xin allows John playing a resource during 2003. Listing 6.11 reproduces verbatim such exemplary license.

Listing 6.11: The first example of REL license

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <r:license
3   xmlns:r="urn:mpeg:mpeg21:2003:01-REL-R-NS"
4   xmlns:sx="urn:mpeg:mpeg21:2003:01-REL-SX-NS"
5   xmlns:mx="urn:mpeg:mpeg21:2003:01-REL-MX-NS"
6   xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
7   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
8   xsi:schemaLocation="urn:mpeg:mpeg21:2003:01-REL-MX-NS
9     rel-mx.xsd">
10   <r:grant>
11     <r:keyHolder licensePartId="John">
12       <r:info>
13         <dsig:KeyValue>
14           <dsig:RSAKeyValue>
15             <dsig:Modulus>KtdToQQyzA==</dsig:Modulus>
16             <dsig:Exponent>AQABAA==</dsig:Exponent>
17           </dsig:RSAKeyValue>
18         </dsig:KeyValue>
19       </r:info>
20     </r:keyHolder>
21     <mx:play/>
22     <mx:diReference>
23       <mx:identifier>urn:grid:a1-abcde-1234567890-f
24     </mx:identifier>
25   </mx:diReference>
26   <r:validityInterval>
27     <r:notBefore>2003-01-01T00:00:00</r:notBefore>
28     <r:notAfter>2004-01-01T00:00:00</r:notAfter>
29   </r:validityInterval>
30 </r:grant>
31 <!--The license is issued by Xin, the distributor.-->
32 <r:issuer>
33   <r:keyHolder licensePartId="Xin">
34     <r:info>
35       <dsig:KeyValue>
36         <dsig:RSAKeyValue>
37           <dsig:Modulus>X0j9q99yzA==</dsig:Modulus>
38           <dsig:Exponent>AQABAA==</dsig:Exponent>
39         </dsig:RSAKeyValue>
40       </dsig:KeyValue>

```

```

41         </r:info>
42     </r:keyHolder>
43     </r:issuer>
44 </r:license>

```

### 6.7.2 Equivalent Permission in core MVCO

This subsection shows the closest information that can be represented with the core MVCO model.

Compared to REL, the time condition (lines 26-29 in Listing 6.11) cannot be written directly with MVCO classes instances. Also, the RSA keys of issuer (lines 35-40) and principal (lines 13-18) have to be omitted. The right `mx:play` in line 21 has to be generalized as a `mvco:Render`. The rest of the information is kept.

MVCO permissions and REL licenses satisfy different purposes, and it has to be noted that additional information is required in the MVCO permission: the statement by which Xin is said to have the rights over the resource. This implicitly gives the permission a certain consistency, being no needed further verifications. The equivalent permission is a piece of OWL, given in Listing 6.12 using the Turtle syntax [39]. The resource (lines 22-25 in Listing 6.11) is here written directly in line 10 of Listing 6.12, but also the MPEG-21 DII element `dii:RelatedIdentifier` could have been used -it is defined in the MVCO.

Listing 6.12: Equivalent core MVCO Permission

```

1  :John rdf:type :User .
2  :Xin rdf:type :User ;
3      :isRightsOwnerOf grid:a1-abcde-1234567890-f .
4  :permission001 rdf:type :Permission ;
5      :permitsAction :render001 .
6      :issuedBy :Xin ;
7  :render001 rdf:type :Render ;
8      :actedBy :John ;
9      :actedOver grid:a1-abcde-1234567890-f .
10 grid:a1-abcde-1234567890-f rdf:type :IPEntity .

```

### 6.7.3 Equivalent Permission with extended MVCO

In this case, first the extensions have to be defined, and then the individuals in the ontology have to be given. The first piece of missing information is the RSAKey. `RSAKeyExponent` and `RSAKeyModulus` are two data type



properties which can be attributed to users and which take string values (see Listing 6.13).

Listing 6.13: Extension of MVCO to describe a User's RSAKey

```
:hasRSAKeyExponent rdf:type owl:DatatypeProperty ;
                    rdfs:domain :User ;
                    rdfs:range xsd:string .
:hasRSAKeyModulus  rdf:type owl:DatatypeProperty ;
                    rdfs:domain :User ;
                    rdfs:range xsd:string
```

These new elements in the ontology can be given a new namespace, *mvcoex*. Xin and John can get now their RSAKey data (Listing 6.14).

Listing 6.14: User class individuals with RSAKey

```
:John rdf:type :User ;
      :hasRSAKeyModulus "KtdToQQyzA==" ;
      :hasRSAKeyExponent "AQABAA==" .
:Xin  rdf:type :User ;
      :hasRSAKeyModulus "X0j9q99yzA==" ;
      :hasRSAKeyExponent "AQABAA==" ;
```

Another possible extension to precise the license's content is subclassing *Play* as a kind of *Render* (Listing 6.15)

Listing 6.15: Play as a new Render subclass

```
:Play rdf:type owl:Class ;
      rdfs:subClassOf :Render .
```

After this, *render001* is declared of type *mvcoex:Render*.

In order to extend the core MVCO to specify a time condition, a new subclass of *Fact* has to be added: the temporal context, which may have an attribute of year. Note that the XML Schema datatypes can be used, like *xsd:gYear* (Listing 6.16).

Listing 6.16: TemporalContext as a new Fact

```
:TemporalContext rdf:type owl:Class ;
                 rdfs:subClassOf :Fact .
:hasYear         rdf:type owl:DatatypeProperty ;
                 rdfs:domain :TemporalContext ;
                 rdfs:range  xsd:gYear .
```

The permission is now given to act only in that year, therefore, *permission001* has required the instantiation of this fact (Listing 6.17).

Listing 6.17: Condition imposed to the permission

```

:year2003 rdf:type :TemporalContext ;
          :hasYear "2003"^^xsd:gYear .
:permission001 rdf:type :Permission ;
              :hasRequired :year2003 .

```

Listing 6.18 shows the complete extension that has to be added to MVCO in order to represent the exemplary REL license.

Listing 6.18: MVCO extension to represent a exemplary REL license

```

:Play rdf:type owl:Class ;
      rdfs:subClassOf :Render
:TemporalContext rdf:type owl:Class ;
                 rdfs:subClassOf :Fact .
:hasYear rdf:type owl:DatatypeProperty ;
         rdfs:domain :TemporalContext ;
         rdfs:range xsd:gYear .

```

Listing 6.19 shows the equivalent assertions in MVCO to codify this particular REL license, to be stored as an individual file or in a database.

Listing 6.19: MVCO assertions equivalent to the exemplary REL license

```

:John rdf:type :User ;
      :hasRSAKeyModulus "KtdToQQyzA==" ;
      :hasRSAKeyExponent "AQABAA==" .
:Xin rdf:type :User ;
     :hasRSAKeyModulus "X0j9q99yzA==" ;
     :hasRSAKeyExponent "AQABAA==" ;
     :isRightsOwnerOf grid:a1-abcde-1234567890-f .
:permission001 rdf:type :Permission ;
              :issuedBy :Xin ;
              :permitsAction :render001 ;
              :hasRequired :year2003 .
:render001 rdf:type :Play ;
           :actedBy :John ;
           :actedOver grid:a1-abcde-1234567890-f .
grid:a1-abcde-1234567890-f rdf:type :IPEntity .
:year2003 rdf:type :TemporalContext ;
          :hasYear "2003"^^xsd:gYear .

```

## Chapter 7

# Agreements representation

This Chapter analyzes a representative set of agreements in the audiovisual sector and extracts its most relevant features deciding which potentially enforceable clauses, rights and conditions are to be taken into account. The MPEG-21 REL license, which can partially represent these contract clauses, is then extended with the necessary new elements. Finally, the whole contract clauses enforceable or not, are represented in an extended eContracts schema with elements of the Media Value Chain Ontology and the audiovisual elements.

### 7.1 Analysis of Contracts in the Value Chain

The analysis done in this Section around contracts has been based on a set of 40 real narrative contracts from the audiovisual market, provided by the AFI<sup>1</sup>, which represents some 200 small and medium enterprises of audiovisual producers. These documents from the real world have a variate provenance, and represent several different contract types, concerning vertically the different steps in the value chain, and horizontally the different media type object of trade (see Table 7.1). They have been also applied in different countries and written in different languages. Other contracts have analyzed from other sources to have a complete view of the value chain, too.

The 40 contracts accounted an average of 8 pages, and 17 clauses each. A selection with the 11 more representative contracts was done based on their main features as shown in Table 7.2. Although clauses are representative, a single clause sometimes represented several complex ideas while sometimes

---

<sup>1</sup>AFI, Associazione dei Fonografici Italiani, <http://www.afi.mi.it/>

<i>Nr. contracts</i>	<i>Material kind</i>	<i>Examples</i>
14	Video	tv programs, films, music clips
11	Audio	music, ringback mobile tones
5	Images	photos, digitalized archives
3	Text	news, ebooks
7	Other	software, multimedia DVD

Table 7.1: Different kind of resources traded in analyzed contracts

just one idea spanned several clauses. Clauses were analyzed and classified, and the following list was extracted as a summary of the main clauses in agreements on audiovisual material.

- **Permissible** Equivalent to ‘The licensee can’

*Rights* The licensee can exercise certain rights. This is usually the first and main clause

*Resource* The referenced resource is either mentioned in the first clause as well, or detailed as an appendix when it is a list of items

*Report and Auditing* In distribution contracts where benefits have to be distributed according to the sales, these sales have to be reported

- **Obligatory** Equivalent to ‘The licensee must’

*Fee* The licensee must pay a fee subject to the described conditions

*Territory* The licensee must exercise the right (if he/she does) in a given location

*Term* The licensee must exercise the right (if he/she does) in the given time frame

- **Impermissible:** Equivalent to ‘The licensee must not’

*Confidentiality* In B2B relations there is usually a clause banning the public issue of information

- **Claims** Equivalent to ‘Something is’

*Disclaimer* To deny responsibilities on certain issues etc.

*Jurisdiction* In case of dispute, the agreed jurisdiction and court is agreed

*Breach and termination* These clauses provision the end of the contract in normal or abnormal conditions

<i>Num.</i>	<i>Kind</i>	<i>Media type</i>	<i>Contract name</i>
2	Online dist.	Music	Online music retail agreement
3	Online dist.	Video	Contract Online Distributor
5	Distribution	Ringback tones	Ring Back Tone License Agreement
10	Distribution	Video excerpts	Video Excerpt License Agreement
13	Online dist.	Images	Image Reproduction Licence
18	View	Images	License agreement for the digitalization and of primary sources
20	Synchronis.	Audio	Synchronisation license from record company
21	Synchronis.	Movie audio	Master use recording license
24	Synchronis.	Audio in advertis.	Licence for use of sound recording in advertisement
28	Online dist.	Music	On line distribution agreement
29	Broadcast	Video	Film clip License

Table 7.2: Features of selected sample contracts

The most important clauses fit into one of these categories: *permissible*, *impermissible*, *obligatory* (all of them subject to the deontic logic) or *claims* (sentences that are considered true). Indeed, not always clauses are easily classifiable in one of those sets, as they do not fall into one of the categories exposed before, and it is not rare finding clauses with double purpose.

Also present in any contract, are the signature of the parties, the date and place of the signature, the title of the contract, a contract reference number etc.

### 7.1.1 Parties in audiovisual contracts and the Value Chain

The *motto* of this Thesis has been that business to business relations can benefit from the same schemata that currently DRM provides in the business to consumer segment if extended adequately. Business to consumer relations usually are regulated by implicit contracts (as with purchases in physical shops) or as end user licenses (as with purchases in electronic shops). In the B2B transactions in the value chain, written contracts (narrative contracts) prevail.

Every contract represents an agreement between two parties who belong to the value chain. Specifically, they are only binded those which have relation according to the intellectual property value chain, and in consequence we can classify the kind of contracts according to the signing parties. Fig. 7.1 shows the typical name of the contract types and relates them with the

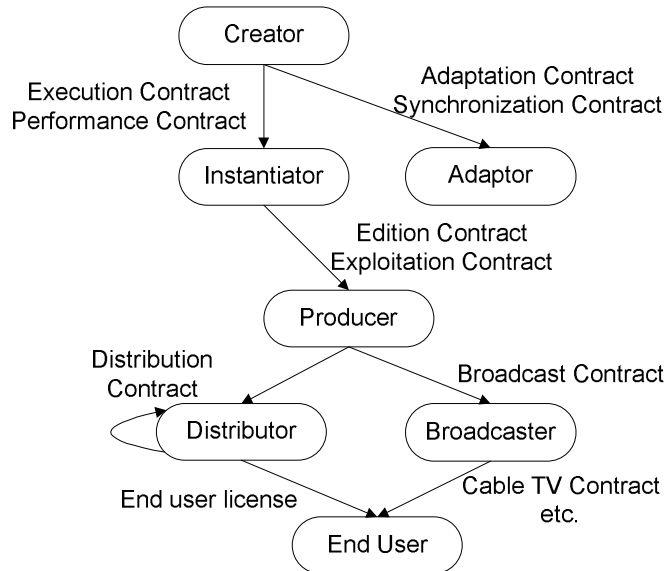


Figure 7.1: Common names of contracts in the Value Chain

parties.

Adaptation contracts are given by the author (or usually his representative) to another artist to perform an adaptation to other instrument, rhythm or similar. The synchronization contract can be used by a film, video or television producer to contract with a songwriter or the songwriter's publisher to use a song in a film, video, music video, television program, or television advertisement whereby the song is synchronized to the action on the screen.

With execution or performance contract, the composer (work's creator) authorises another performer artist to execute publicly the work (in a concert, on a theater play etc.). Distribution contracts allow the producer to produce and distribute the interpreted work (instance) between retailers, service providers, broadcasters etc. The later relations are expressed with redistribution contracts or broadcasting contracts.

For these contracts to be executed automatically in a digital platform, there is a lack of expressivity in current Rights Expression Languages, which so far focused only in elements for the distributor to end user stage. The next Section describes which elements are needed in particular, based on a broad contract analysis.

Table 7.3 shows the distribution of contract types found in the collection of analyzed contracts according to the parties' nature.

<i>Nr. contracts</i>	<i>Contract type</i>	<i>Subtypes</i>
28	Distribution	Traditional, online
6	Synchronization	Audio in video, images in video
7	Broadcasting	Satellite, cable, internet streaming
5	Download	Music, images, text
4	Edition	News in newspaper, images in website

Table 7.3: Different contract categories in the set

### 7.1.2 Rights in audiovisual contracts

The task of deciding which rights are being used in which contracts is not a trivial one. In fact, every REL or DRM system tries to declare its list of rights, with an English definition (see Table 5.2 or Table 5.4). This is needed to share a common vocabulary. Contracts often have an appendix with the definitions as agreed by both parties, reducing thus ambiguity. After the process of analysis, jointly made with AFI, the rights shown in Tables 7.4 and 7.5 were deemed as the most relevant, and the definitions attached as the most accurate.

As an example of procedure, and counting also with the context in which they were written, the following clauses were simplified with the right 'broadcast' (brackets show the contract number, clause and subclause where the text appears):

- 6.2 [*Licensor hereby grants to LICENSEE the*] *right and license under copyright to broadcast, exhibit and/or display any and all versions of the Pictures*
- 6.2.1 [*Licensor hereby grants to LICENSEE the*] *right to distribute and publish the Pictures*
- 21.7.a *To utilize such Video Records for any and all purposes uses and performances*
- 7.2.1.i *BROADCASTER grants to Web company the non-exclusive right and license to use, copy, publicly display, publicly perform, distribute, or otherwise make the BROADCASTER Content available on the Web company*

### 7.1.3 Conditions in audiovisual contracts

The analysis of the contracts has allowed to identify a list of rights which can be considered as typically used in the license contracts. The same procedure can be done with the clauses representing conditions or prohibitions. However, the conditions of the contract are more shaped by the negotiation between the parties, they vary more from contract to contract and therefore drawing a complete list is more difficult.

reproduce	to authorize the act of rendering the content in any manner or form (i.e. reproduction covers all methods of reproduction for instance drawing, lithography and other printing processes, photocopying, recording)
download	to copy data from a main source to a peripheral device
upload	to transfer data from a peripheral computer or device to a central computer
make available	the "posting" or storage of material or information on a computer or server connected to the World-Wide-Web or connection of a computer containing material or information for access using the Internet or an intranet
publicly perform	presenting or executing the work in a place open to the public or at a place where a substantial number of persons outside of a normal circle of a family or social acquaintances are gathered
exhibit	to show outwardly
transmit	to send data over a communications line
broadcast	to send out or communicate, especially by radio or television
copy	to manipulate the licensed content in order to produce a new digital object whose characteristics are the same as the original one and which is autonomous from the latter

Table 7.4: Rights found in contracts(I)

Nevertheless some of them are truly recurrent, and those which appear in Table 7.6 are quite representative.

## 7.2 MPEG-21 REL for representing audiovisual contracts

Considering the role that REL licenses play on DRM systems, RELs can be seen as effective electronic contracts that are being enforced. Contracts usually express reciprocal rights and obligations, but in the framework of the audiovisual sector, most of the clauses refer to the rights of one of the parties and the related conditions to be met upon exercise. This model fits well with the information structure conveyed in an MPEG-21 REL license, where rights and conditions lie in the same side.

This section evaluates the MPEG-21 REL licenses for representing audiovisual contracts, and extends it as needed to perform this task. New elements to be inserted in REL licenses will be defined in order to complete the MPEG-21 REL.



publish	to prepare and issue for public distribution or sale
print	to produce something in printed form by means of a printing press or other reproduction process
record	to register (sound or images) in permanent form by mechanical, electrical or electronic means for reproduction
modify	to change in form or character
translate	to render in another language
dub	To insert a new soundtrack, often a synchronized translation of the original dialogue.
adapt/edit	to make suitable to or fit for a specific use or situation
convert	to change a content into another format
transcode	transcoding is the direct digital-to-digital conversion from one to another.
remix	to recombine (audio tracks or channels from a recording) to produce a new or modified audio recording
distribute	to supply contents to retailers
sell	to exchange or deliver for money or its equivalent.
lease	to grant the right of possession and use of a content for a specified period in exchange for payments
promote	to attempt to sell or popularize by advertising or publicity
synchronize	to cause (soundtrack and action) to match exactly in a film
license	to grant a license to or for; authorize
sub-license	to license again all or just some of the rights which have been granted to him.

Table 7.5: Rights found in contracts (II)

Term	period of time during which the conditions of a contract will be carried out.
Territory	the area where the rights granted might be performed
Exclusivity	regulates if one party grants another party sole rights
Fee	the clause (or the clauses) which regulate this aspect are aimed at disciplining the remuneration of the licensor
Reporting	information to be given, linked to the need of the licensor of monitoring the use of the content in order to set the amount of the remuneration

Table 7.6: Some conditions found in contracts

### 7.2.1 Mapping of contract clauses to license elements

After the analysis of the previous section, which summarizes the main clauses present in audiovisual contracts, a matching can be done to the existing MPEG-21 REL elements (and presented in Section 5.3.1).

#### Metadata mapping

Metadata in the contract are the prologue and epilogue sections in a contract which specify the title, date, the signatures, the identification number or reference of the contract, the author of the contract, etc. Mapping to the MPEG-21 REL license is quite straightforward.

Contract title maps to the `r:title` element in the REL license, contract identification number maps to the `r:id` attribute of the `r:license` element. Handwritten signature is mapped as a digital signature in the `dsig:Signature` element. Date in the contract is mapped into a `dc:date` element in the `r:otherInfo` REL element, as well as the rest of the metadata attributes like the place of the signature (`dc:location`), the writer of the contract etc.

#### Parties mapping

MPEG-21 REL licenses refer always to two parties, namely, the issuer and the principal (actually a REL license may include several grants each of them with a different party, but then we can consider the grant as the basic license unit). The identification with the contract parties is immediate.

In MPEG-21 REL no more information is given about who might be these parties, excepting that they are uniquely identified, and that one of them (the rights granter) electronically signs the document.

In the MPEG-21 Multimedia Framework, User is defined as

*... any entity that interacts in the MPEG-21 environment or makes use of Digital Items, including individuals, consumers, communities, organizations, corporations, consortia, and governments and other standards bodies and initiatives around the world...*

According to the standard, users are only defined by the actions they perform, but if we attend to the expressivity of the REL, in the licenses there can be only end users and distributors. We can classify the party receiving the right from the right that is granted (Table 7.7).

<i>right</i>	<i>party</i>	<i>right</i>	<i>party</i>
issue	distributor	extract	end-user
revoke	distributor	embed	end-user
possessproperty	end-user	play	end-user
obtain	distributor	print	end-user
modify	end-user	execute	end-user
enlarge	end-user	install	end-user
reduce	end-user	uninstall	end-user
move	end-user	delete	end-user
adapt	end-user		

Table 7.7: Classification of parties from the rights

MPEG-21 REL does not characterize in depth more types of users than end users and distributor but a contract model should consider all the roles appearing in Fig. 7.1. Users should be characterized with the roles of the MVCO.

### Rights mapping

Table 7.8 lists the most important rights granted by the licensor to the licensee and put in evidence by the analysis of the contracts. A right is extracted from an English sentence as the assertive words by which one party concedes a privilege to the other having removed objects and conditions.

The table includes the rights found in the narrative contracts in the first column and the MPEG-21 REL equivalent term next to it. When no MPEG-21 REL equivalent right has been found, a new element has been created with the namespace `avc`. This `avc` prefix stands for *audiovisual contracts extension*.

### Conditions mapping

Term and territory are two of the most frequent conditions that appear in contracts, and they are easily mappable. For example, extracted from contract, this clause if found in the contract:

*Licensor grants to Licensee the exclusive right, privilege and license, [...] throughout the Territory of the People's Republic of China"*

This would be encoded in MPEG-21 REL as shown in Listing 7.1, provided that the namespaces are established as `xmlns:sx="urn:mpeg:-`

<i>Right</i>	<i>MPEG-21 REL</i>
To reproduce	avc: reproduce
To download	m1x:governedCopy
To upload	m1x:governedCopy
To make available	r:issue
To (publicly) perform	avc:perform
To exhibit	avc:perform
To transmit	avc:transmit
To broadcast	avc:broadcast
To copy	m1x:governedCopy
To publish	avc:publish
To print	mx:print
To record	avc:record
To modify	mx:modify
To translate	avc:translate
To dub	avc:dub
To change	mx:adapt
To edit	mx:adapt
To convert	mx:adapt or mx:modify (with constraints)
To transcode	mx:adapt or mx:modify (with constraints)
To remix	avc:remix
To distribute	r: issue
To sell	r:issue (sx:FeeFlat + sx:FeeMetered + sx:FeePerInterval + sx:FeePerUse + sx:FeePerUsePrePay)
To lease	Right + Payment Condition + validityInterval
To synchronize	avc:synchronize
To license	r:issue r:delegationControl
To sub-license	r:issue r:delegationControl
To promote	avc:promote

Table 7.8: Mapping of rights from contracts to MPEG-21 REL

mpeg21:2003:01-REL-SX-NS" and xmlns:iso="urn:mpeg:mpeg21:2003:-01-REL-SX-NS:country".

Listing 7.1: Example of condition mapping

```
<sx:territory>
  <sx:location>
    <sx:country>iso:CN</sx:country>
  </sx:location>
</sx:territory>
```

In another example, we find a clause in Italian language:

*Per la durata del presente accordo, la licenziante conferisce a X che accetta, il diritto esclusivo di [...] nei territori di: Italia - Citta' del Vaticano - Repubblica di San Marino"*

What would be codified in the same but using the elements: `iso:SM`, `iso:VA`, and `iso:IT` (the standard ISO3166 for regional codes is used). We note that regardless the human language and on despite of using different words, the meaning is the same and the codification into the electronic license is valid. Not always the mapping is so clear, and sometimes the territories are given implicitly. For example, the following clause:

*The territory in which Licensor may exercise each and all of the rights granted herein shall be the territory of North, Central and South America ('Territory')*

there should come an enumeration with all the countries of the American continent. This interpretation of the contract cannot be easily performed by a computer in an automated way, but once translated into the digital license language, the computer could easily enforce the fulfillment of the condition.

The three previous contracts referred to three different kinds of audiovisual material, the first contract was with ringback tones, the second with television material and the third with motion pictures. Note the abstraction that is made, applying the same concept to different resources and still keeping the consistence.

## 7.2.2 Extension of MPEG-21 REL to represent audiovisual contracts

### Extension of parties

Parties types can be further specified in the user roles defined in the MVCO. An XML Schema can be extracted in a manner that for each of the user

role classes (e.g. most of `mvco:User` subclasses) can qualify the users in the MPEG-21 REL schema.

### Extension of the resource

In MPEG-21 REL licenses resources are given usually by a DII reference but its nature is not specified. The intellectual property character of the pointed resource can be further precised with the use of the MVCO `mvco:IPEntity` subclasses (e.g. `mvco:Work`, `mvco:Manifestation`, etc.).

### Extension of rights

Table 7.8 anticipated that not all the rights appearing in contracts could be mapped to the MPEG-21 REL rights. Those rights that were missing were labeled under the namespace `avc`. This constitutes the first extension of the elements provided by MPEG-21 REL.

Very often the rights granted are given with an exclusivity character. This is something not easy to handle within MPEG-21 REL, but some workarounds are possible [156].

### Extension of conditions

Also some of the conditions analyzed in previous Sections could not be represented with current MPEG-21 REL elements (nor from its profiles). The following new conditions are required for expressing restrictions in audiovisual contracts.

The `avc:validityPeriodRenew` condition (Fig. 7.2) is needed to express that after a given period, if none of the parties communicates so, the contract will be automatically renewed. It consists of three child elements: `avc:-start`, `avc:duration` and `sx:feePerInterval`.

The element `avc:start` indicates the date from which the period designated in this condition becomes meaningful. This element is optional because the issuer of the license cannot know the starting date of the period. The `avc:duration` element indicates the period of time during which rights can be exercised. Finally, the `sx:feePerInterval` determines the payment for renewing the rights and its `r:serviceReference` element controls the days that will be renewed.

A `avc:nonTerritory` condition element (see Fig. 7.3) identifies the country in which rights *cannot* be exercised. Then, rights are granted for the entire world except for a single country or region.

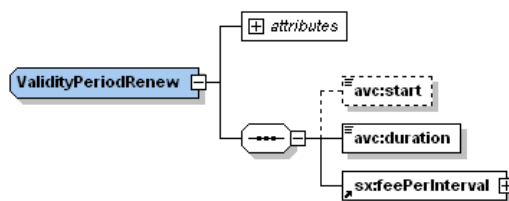


Figure 7.2: validityPeriodRenew element

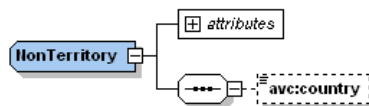


Figure 7.3: nonTerritory element

A `avc:interestFee` condition element (see Fig. 7.4) is used to express an interest fee to be delivered in case of payment delay. It consists of three child elements, the `avc:percentage` element determines the percentage of interests that shall be paid, the `avc:from` element specifies the starting date from which the interests shall be paid and finally, the `avc:to` element defines to which entity shall be paid the interests.

The `sx:feeFlat` condition, present in the standard MPEG-21 REL extension, has been extended with the `avc:paymentType` element to allow specifying in this condition how the remuneration will be paid, for example in cash, or in stocks of a company etc. The new `avc:feeFlatEx` is shown in Fig. 7.5.

The new rights and conditions defined in this section were gathered in a XML Schema and precisely described in [157].

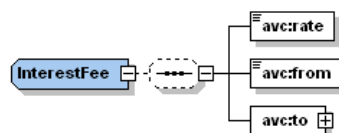


Figure 7.4: interestFee element

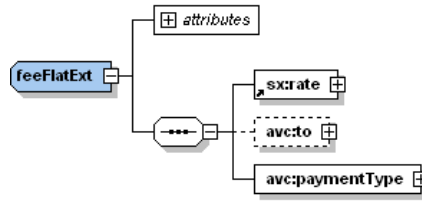


Figure 7.5: feeFlatExt element

Kind of clause	Meaning	Example clause
<code>avc:permission</code>	What the licensee can do	Licensee rights
<code>avc:prohibition</code>	What the licensee cannot do	Confidentiality
<code>avc:obligation</code>	What the licensee must do	Fee, territory, term
<code>avc:assertion</code>	What both parties agree it is	Jurisdiction
→ <code>ec:metadata</code>	Data on the contract itself	Contract date
→ <code>ec:contract-front</code>	Contract heading	Contract parties

Table 7.9: Clauses classification in eContracts extension

### 7.3 eContracts representation for audiovisual contracts along the value chain

As it was reviewed in Section 4.1.5, eContracts is a standard aimed at representing general contracts, having no particular specialization as scope. As it was seen, eContracts documents were composed of general paragraphs and clauses, being the main XML elements the `ec:item`, the `ec:title`, the `ec:block` and the `ec:text` with the `item` element used recursively (`ec` is the prefix for the namespace of eContracts). But these fields can be extended, being the contract parties declared in the `ec:contract-front` part, and the clauses under the `ec:body` element.

Instead of declaring the kinds of clauses (those listed in pag. 140) as simple descendants of `ec:item`, they can be better classified to belong to one of the standard deontic logics notions of ‘prohibition’, ‘obligation’, ‘permission’ or ‘assertion’ aforementioned depending of the meaning they convey. These four categories can be made derived from `ec:item`. Table 7.9 shows some examples, where the namespace referenced by `avc` stands for audiovisual electronic contracts schema. Contract metadata and in contract front constitute assertions.

Parties in the contract front can be further specified with the elements of the Media Value Chain Ontology; listing 7.2 shows a fragment of where the names Alice and Bob are given (in a real scenario, these names would



be URI identifiers). Note the use of the MVCO elements (*Distributor* and *EndUser*), which are not directly included in the OWL ontology itself but which can be directly obtained from the OWL classes-to-XML mapping.

Listing 7.2: Contract front with MVCO parties

```
<ec:contract xmlns="urn:oasis:names:tc:eContracts:1:0">
  <ec:contract-front>
    <ec:parties>
      <ec:party><mvco:Distributor rdf:about="#Alice"/></ec:party>
      <ec:party><mvco:EndUser rdf:about="#Bob"/></ec:party>
    </ec:parties>
  </ec:contract-front>
</ec:contract>
```

The XML snippet in Listing 7.3 asserts that there is a work called *mywork1* (lines 16-18), that there is a Fact called *Germany* (lines 13-15), and that there is an action that is *Bob* ‘making an adaptation over mywork1’ (lines 09-12). It also declares that there is a permission given by Alice for Bob to make an adaptation provided that it is in Germany (lines 03-07). All this information is enforceable (line 02).

Listing 7.3: eContracts clause integrated with a MVCO Permission

```
1 <ec:body>
2 <ec:item>
3 <avc:enforceable>
4 <mvco:Permission rdf:about="#Permission000">
5 <mvco:permitsAction rdf:resource="#Action000"/>
6 <mvco:issuedBy rdf:resource="#Alice"/>
7 <mvco:hasRequired rdf:resource="#Germany"/>
8 </mvco:Permission>
9 <avc:assertion>
10 <mvco:MakeAdaptation rdf:about="#Action000">
11 <mvco:actedBy rdf:resource="#Bob"/>
12 <mvco:actedOver rdf:resource="#mywork1"/>
13 </mvco:MakeAdaptation>
14 <avc:Territory rdf:about="#Germany">
15 <avc:hasCountry>ISO:DE</mvco:hasCountry>
16 </avc:Territory>
17 <mvco:Work rdf:about="#mywork1">
18 <mvco:hasRightsOwner rdf:resource="#Alice"/>
19 </mvco:Work>
20 </avc:assertion>
21 </avc:enforceable>
22 </ec:item>
23 </ec:body>
```

### 7.3.1 Enforceable clauses with the MVCO permission model

For clarity, Listing 7.3 showed the deontic logic expressions with an own element. However, these elements are redundant, only present for an easy processing, given that the same information can be given by the powerful permission model of the MVCO.

OWL DL provides a set of rich operators, as it was seen in Section 3.3 (e.g.  $\neg$ ,  $\vee$ ,  $\wedge$ ,  $\equiv$ ,  $\forall$  etc.) but it does not provide the deontic operators of obligation nor necessity ( $\sharp$ ,  $\diamond$ ). With the obligation operator  $\sharp$ , it is possible to express the obligations ( $\sharp P$ ), the permissions ( $\neg\sharp\neg P$ ) or the prohibitions ( $\sharp\neg P$ ).

One of the ways of emulating the operator in the model is introducing one object property linking conditional facts for something to happen, i.e. a fact that must hold or must not hold for a permission to be valid etc. This object property could have been either `isPermissible` or `isObligatory` (there is a duality in the operators).

The MVCO declares the second as the OWL object property `mvco:-hasRequired`. A `Permission` may `mvco:hasRequired` zero or more `mvco:-Fact` to hold. However, by merely adding the object property `mvco:-hasRequired` and the two Kripke axioms (Section 3.1.3), it is not possible to represent deontic propositions and still stay within the limits of OWL DL and its attractive tractability. The negation operator in OWL, `owl:-complementOf`, cannot be used for classes in an ontology and still remain within the OWL DL, it can only be applied to class individuals. The easiest solution is to simply define the negated operation, permission, as another object property. In the MVCO, the object property is called `mvco:-permitsAction`. Both object properties can be seen in Fig. 6.7. Note that in the MVCO, these object properties relate the permission with an action and a fact, but nothing prevents actions to be considered as facts (for example, disclosing a certain content can be forbidden or can be mandatory; in which case the hypothetical extended element `mvcoex:disclose` would subclass the `mvco:Fact` and the `mvco:Action` at the same time. Therefore, `mvco:-hasRequired` and `mvco:permitsAction` fully allow expressing the complete set of expressions of the deontic logic.

An authorisation is the decision of validating a certain request to execute an action (an instance of `mvco:Action` and associated classes), given a certain `Permission` (an instance of `mvco:Permission` and associated classes), and a certain context (individuals of `mvco:Fact` that hold in a given moment).

The reasoning power of this formalisation can be employed in different

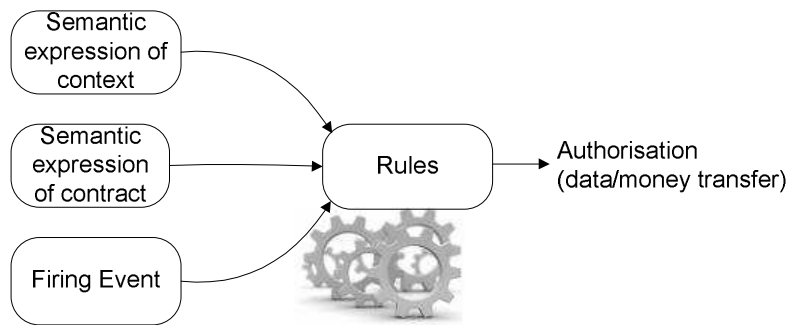


Figure 7.6: Semantic authorisation

manners to effectively perform the authorisation. The fastest is applying a consistency test. For example, if the permission does not exist, or if it is given by an agent not being the rights owner, the consistency test will promptly fail and the request will be denied. However, in order to validate more complex constraints, like the conditions posed through `mvco:-hasRequired mvco:Fact`, the semantic web rules can be used. For this, first the context has to be transformed into instances of `mvco:Facts`, as shown in Fig. 7.6.

In Fig. 7.6, a firing event (for example but not only an authorisation request) starts the rule execution, which taking the contract (as one or more instances of `mvco:Permission`) and the context (set of `mvco:Facts`) changes the value of a boolean data type property (`mvcoex:isAuthorised`) which can be added to the `mvco:Action` request. Action class individuals would belong then to one of the following categories: executed actions, actions created to express authorisations, and request actions. The high expressivity of possible conditions, obligations and bans probably compensates the (low) speed of execution of the authorisation process.



## Chapter 8

# Implementation results

This Chapter presents the practical results associated with the work presented in the previous Chapters. The ideas of this Thesis are supported by its materialization in several related projects.

While engineering projects have a fixed duration and some of the projects described here have actually already finished, the goal of this Chapter is to demonstrate the use of the ideas exposed in the previous chapters and the main thesis: that the intellectual property value chain can have a precise, formal representation able to arbitrate e-commerce relations around audiovisual content and spanning each of the chain links.

Three of the projects (AX4HOME, E2E, MXM) show the ontology use in the center of a DRM framework; AXMEDIS shows how contracts can be represented as licenses and finally VISNET-II shows how the ontology can authorize diverse operations of minor importance. Finally, a common element in these projects will be described: the ontology APIs.

### 8.1 The Ontology in the project AXMEDIS

#### 8.1.1 Introduction

##### Goals

The AXMEDIS DRM Platform was presented in Section 5.1.1. An innovative goal of the AXMEDIS project and beyond the state of the art, established that REL licenses could be issued as the result of the semi-automatic translation from existing narrative contracts. Companies in the media distribution market usually spend a significant amount of time in narrative

contracts management. This time could be reduced if an electronic version of contracts existed. The ideal goal would be to *replace the existing narrative contract by its associated digital license in an automated process*.

Regarding automated translation, current technology imposes some limits. Relationships between narrative contracts and digital licenses, involve human language processing, what is known between the scientific community as *natural language processing*. This is a non mature branch of artificial intelligence, that with the given state of the art can only provide results of a limited quality, and while imprecise results might still be of interest in some applications, wherever legal issues are involved, accuracy is a key attribute. Therefore, the fully automatic transformation was discarded and then a supervised process was proposed instead.

On the other hand, a complete representation of the information of a real life narrative contract in terms of a machine oriented language is never possible; a representation always loses fidelity from the represented thing (the assert is trivial, but cannot be forgotten, especially because of legal issues where nuances are crucial).

However, once admitted that a digital license can represent a limited range of situations always narrower than a paper contract, the digitalisation can still be useful for more modest goals, and so was it assumed in the AXMEDIS project. The goal could then be formulated as *semi-automatic processing of the essentials of a contract*.

### Contract management models

Having declared the goal, two different models were proposed.

1. The first did not aim at replacing narrative contract but complementing them. Thus, the contract would exist as it was having worthy validity under the legal courts, but a license with the core information would be derived (semi-automatically) from the contract, having validity for the computer systems (i.e. B2C distribution of material etc.).

2. A second model was proposed, also without making contracts to disappear but at least to digitalise them to some extent. Contracts would remain, but with a different structure where representable clauses were well separated from those which were not. Those non representable clauses would still be part of the contract, but in a clearly separated section.

In this second model, digitalization could be done much faster, if following a computer guided process. Template clauses would be stored in the computer system, and a contract redaction would be a mere guided process (an assistant like operation). This digitalization of contracts (not merely

digitalization of the texts, but also of its contents) would bring undoubtedly benefits, at the expense of rewriting the existing contracts:

- Contract databases could be created
- Search in a contracts database would be much faster
- Reports and analysis of results based on this type of contracts could be now automated tasks (i.e. it could be shown which contracts turn out to be profitable etc.)
- Generation of new contracts would be much faster.

### 8.1.2 Semi-automatic extraction of licenses from contracts

#### Process description

The process is described in Fig. 8.1. The starting point is the narrative contract, which after an automatic analysis derives in a structured contract. This result is corrected by a human expert, which produces the revised structured contract exportable into the XML eContracts format, or produces an MPEG-21 audiovisual contracts license. From it, again an automated tool can generate text to produce a computer generated narrative contract, textual feedback for an expert to validate the process. License and eContracts are grayed in Fig. 8.1 in order to highlight the products of this process.

This process, if fully automatic, would be subject to a high number of errors, and given its importance, would demand the human supervision in any case. This was not much different from a totally manual operation (as in the existing AXMEDIS DRM Editor and Viewer Tool<sup>1</sup>). The intermediate approach would consist of a pleasant graphical user interface that would be already filled in by the computer (in his guess of what it could find out) but anyway free to be changed by the user manually.

The license extraction is divided in two stages: structure parsing and content parsing.

#### Structural parsing

The first parser separates paragraphs and tries to guess the object of each paragraph. Each of the basic elements (issuer, principal, condition, right...) has a set of keywords that are scanned in each paragraph. The existence or absence of these words can determine which kind of information is contained

---

<sup>1</sup>AXMEDIS Tools can be downloaded from <http://www.axmedis.org/com/>

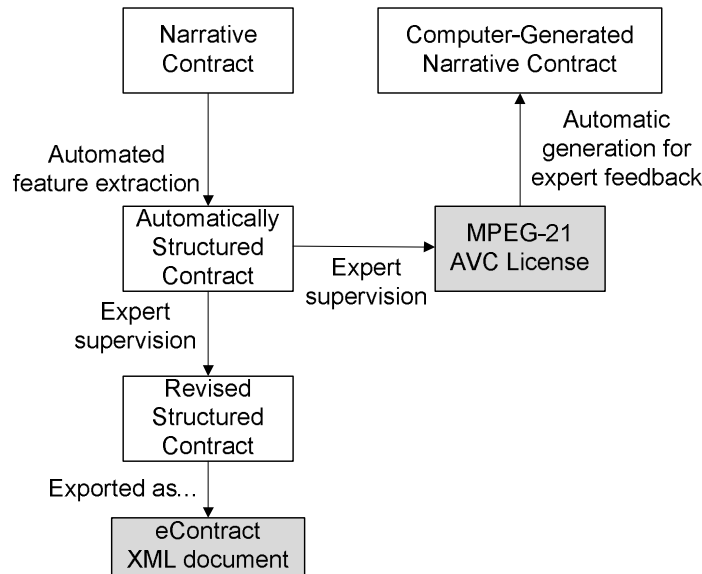


Figure 8.1: Semi-automatic license generation from contracts.

in the paragraph. This analysis bases its decision in a preloaded database, where each of the considered rights and conditions are associated to a set of typical English words, keywords that when analyzing the particular contract will be searched. For example, the ‘territory’ clauses, usually include terms such as ‘country’, ‘territory’, ‘region’ or ‘world’ etc. Each of these words receives consideration, and when analyzing the text contract, an optimal decision will be taken. The results are left in a file is structured as an XML file to be the input of the next stage (following the eContracts Schema).

### Content parsing

The content parser tries to extract then a concrete value. For example, once identified that the clause was of the type ‘territory’, the name of a country is searched. Again, a high rate of mistakes is expected as strings such as ‘North America’ can be hardly identified as Mexico, USA and Canada) by an automatic processor.

The mere identification of the parts in the tagged contract, either in eContracts style or any other, is already an important step that would justify by itself the process of conversion from plain text files to the XML document. It allows a better organized storage of the documents in a contracts database and facilitates their management. But in order to allow



the automatic enforcement of the contracts, a step further must be done and some clause meanings have to be accessible by the computer. Hence a MPEG-21 REL license is generated.

This is done in a guided process, where the user is asked to fill in some forms. The web application will offer sequentially a tentative interpretation of the clauses, that the user will have to confirm or modify the proposed MPEG-21 REL term.

Both the rights and the conditions listed in the audiovisual extension are supported. While this schema works well with some conditions (fee, territory and date), where the vocabulary is rather closed, in other kind of clauses the system may fail to provide a valid suggestion and the user would have to introduce entirely the details.

### 8.1.3 Synthesis of narrative contracts from licenses

For the verification of the generated license, the automatic synthesis of new narrative contracts from the generated licenses can prove to be useful. If the process is fine, the results should be easily verifiable if matching the original contract. This facilitates non-technical staff to give feedback about the transformation quality.

The synthesis of the natural language is much easier. Natural language generation often is characterized as a process that starts from the communicative goals of the license, and then employs some sort of planning to progressively convert them into the written contract. The planning structures first the text, then (for this particular case) a set of clauses is chosen to be written, and finally in the lowest level, words are chosen. It is a long process, that if well done, should define a grammar, a set of rules, a set of words etc., so we can say that is far beyond of our scope.

An easier approach would be a template-based one. Template systems rely on the application of pre-defined templates (i.e. distributor contract) or schemata and are able to support flexible alterations. The template approach is used mainly for multi-sentence generation, particularly in applications whose texts are fairly regular in structure, therefore should be suitable in the context of contract generation.

The contract is automatically structured after a tentative clause identification.

## 8.2 The Ontology in the project AX4HOME

The AX4HOME was a project carried out in order to show how the AXMEDIS framework could be deployed to support the distribution of AXMEDIS rights managed content in particular scenarios: over the internet, on mobile distribution channels and particularly through broadcasting. AX4HOME saw the Value Chain ontology with a more active role in the DRM platform and it was put into practice in several demonstrators.

### The Value Chain Ontology in AX4HOME

The AxIPOntology was the semantic expression of the Intellectual Property Model in the AXMEDIS project along its value chain. Its application took place within a more particular scenario, with more defined conditions. This scenario was the AXMEDIS-4HOME or AX4HOME initiative, which was the result of implementing AXMEDIS for a home, office and mobile environment scenario. It was made up of a number of components from the AXMEDIS framework plus new extensions, being one of the extensions the AxIPOntology.

AX4HOME included scenarios of DRM license creation and user actions authorization among others, where the ontology took an important role. The following uses were identified:

- *In the creation model.* At content creation time, potential available rights were embedded in the object (AXMEDIS object). The ontology was queried to assert that the user effectively had been registered as a creator, and should be informed that such a user had effectively created such object (when sending the object to the database).
- *At licensing time.* When creating a license, either with the *DRM Editor and Viewer* application or the *ContractManager* application, the Ontology had to be checked and informed. It was queried to know if the license was valid according to the relations defined in the ontology, and informed if the license was finally uploaded.
- *At consumption time.* When exercising certain rights, a check was done against the ontology to assure the validity of the operation.

The implementation result of the ontology can be evaluated as far more important than a mere software module; it was a basic strategic keystone establishing the foundations of the Intellectual Property Model, upon which the legitimacy base of the income in any media business was granted. The

ontology was a neutral element, which did not support any business model in particular, but rather it supported all of the them.

In practice, and descending from the abstraction, the AxIPOntology became real as an OWL Ontology, with a set of limited statements describing the elements of the model and their relations. This model was made public and accessible for verification<sup>2</sup>. Furthermore and beyond modeling aspects, practical functions could be built on and/or supported by the ontology.

The most important functionality the ontology could offer was that of certifying that operations of AX4HOME modules were conformant to the agreed IP model.

### Implementation details

According to what had been specified, an implementation was done, resulting in the following elements:

- The AXIPOntology, as a server

The ontology AxIPOntology itself, an OWL file.

A Java API for an easy access to the ontology

A command-line monitor tool, to operate and make basic operations directly on the ontology.

A windows monitor tool, to register and monitor the activity of the ontology by means of tracing the transactions.

- A demonstrator application to test the application

Integration of ontology queries in the ContractManager tool. This software was implemented, tested, and demonstrated in a review in October 2007 in Florence. Fig. 8.2 is useful for recalling the arrangement.

## 8.3 The Ontology in the E2E project

The ‘End to End Rights Management’ project (E2E for short) was a Spanish national project aimed at experimenting the exchange of contents in a

---

<sup>2</sup>*Axmedis DE12.1.4.1. AXMEDIS-4HOME Component Integration, Prototypes and Documentation*, March 2008, is a public document available at <http://www.axmedis.org/-documenti/view.documenti.php?doc.id=3939>

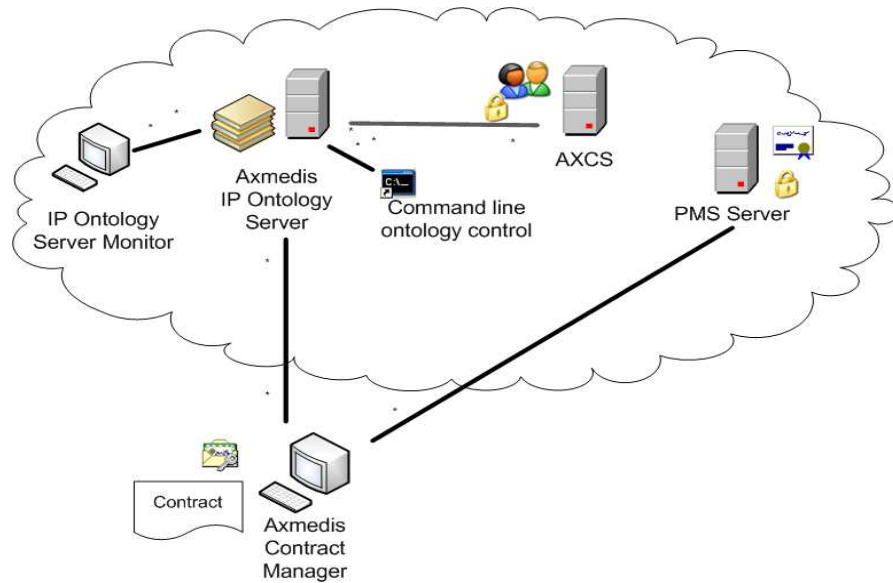


Figure 8.2: The Media Value Chain within the AX4HOME project.

protected environment, with the participation of the Spanish Authors Collecting Society, some universities and some relevant industrial partners (like Telefónica)<sup>3</sup>.

This safe environment was named *Entorno Operativo Común* (EOC), and it shared some features with those described for the AXMEDIS project: an MPEG-21 based solution, with identified users or devices, defined creation tools, content encryption elements, authorisation modules and even software players. On the contrary of AXMEDIS, focus was on interoperation of different commercial platforms already existing. These platforms had different user identification systems, and identity federation was a research priority. Also unlike AXMEDIS, individual users were not allowed to access the platform directly and complex domain management systems were foreseen.

One of the elements in this framework was called *Servidor Semántico* (semantic server), which was no more and no less than the Value Chain ontology with its features adapted to those of the project, and the needed software on top to interact with the other elements in the EOC. The role of the semantic server in the E2E project can be seen in Fig. 8.3 (shadowed).

<sup>3</sup>E2E was a project supported by the Ministerio de Industria, Turismo y Comercio, FIT-350503-2007-9

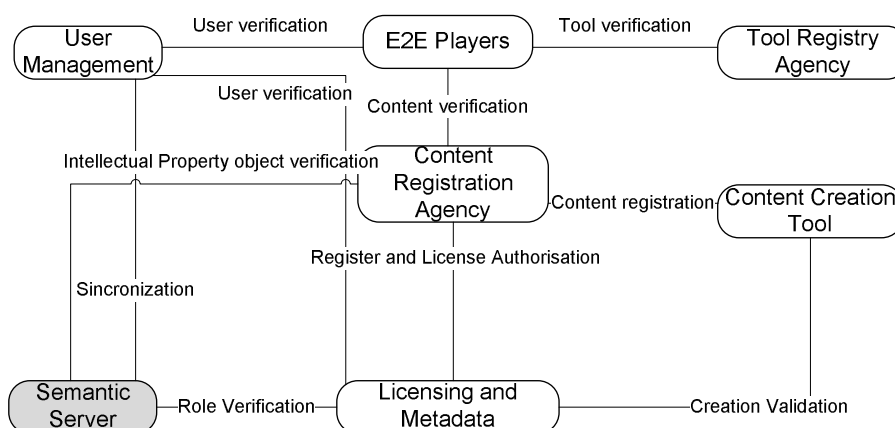


Figure 8.3: Common Operative Environment (EOC)

The semantic server (shaded in the Figure) is synchronized with the user management module, and upon a new registration or an unsubscription in the system the semantic server is informed and updated. Similar communication exists with the content registry agency, which is in charge of receiving new content information. The intellectual property origin of the registered content is checked and validated in the ontology, which is essential for the case of derived content creation (i.e. if an adaptation has to be registered, or the performance in a concert of certain music etc.). The licensing module also verifies the role of the users against the semantic server before an actual authorization is done, as well as the nature of the authorized resource, granting thus the maximum respect to the legal prescriptions.

Communication through Web Services with these modules allows this architecture to be distributed, and to make heterogeneous information systems to interoperate (e.g. those of content providers who inject content, those of service providers which authorize licenses etc.).

## 8.4 The Ontology in the project VISNET-II

### Introduction

VISNET-II<sup>4</sup> was a project partially supported by the European Commission IST FP6 programme. It focused on audiovisual media technologies covering

<sup>4</sup>VISNET-II Networked Audiovisual Media Technologies, Network of Excellence, IST-2005.2.41.5

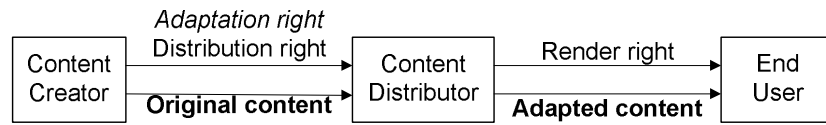


Figure 8.4: Adaptations need to be authorized

areas like video coding, audiovisual media processing and security.

One of the VISNET-II aims was the Virtual Collaboration Scenario, a proposal which allowed remotely located partners to meet in a virtual environment using the state of the art communication and digital multimedia technologies. In such a situation, media had to be delivered to different users, each of them under possibly different conditions: disparate terminal, heterogeneous network quality and varying user preferences. These environment conditions are called also the *context*. The process of converting the media in relation to the context and user preferences is known as *adaptation*.

### Problem statement

However, content is often considered an artistic work and therefore subject to the intellectual property enacted legislation. In the Berne Convention, article 12, it can be read that *Authors of literary or artistic works shall enjoy the exclusive right of authorizing adaptations, arrangements and other alterations of their works*.

If an adaptation is performed without the consent of the rights holder, copyright laws are infringed. Both Content Creators and Content Distributors have to express digitally their authorization in the framework of a Digital Rights Management (DRM) system.

Fig. 8.4 shows the traditional content flow (original content from content creator or performer to the distributor, and once adapted from here to the end user), and the two elemental rights of distribution and right of rendering. But additionally, in italics, it is needed that the content creator authorizes any modification of his work; which is the case when an adaptation is to be performed.

Content adaptation is covered by MPEG-21 Part 7 (Digital Item Adaptation, DIA), and this was the adopted solution in VISNET-II. Digital Item Adaptation identifies and represents context to assist operations to adapt the content under the form of a Digital Item. However, although DIA belongs to the same standard as MPEG-21 REL, there is no explicit manner to make them work together in order to authorize the adaptations. A joint

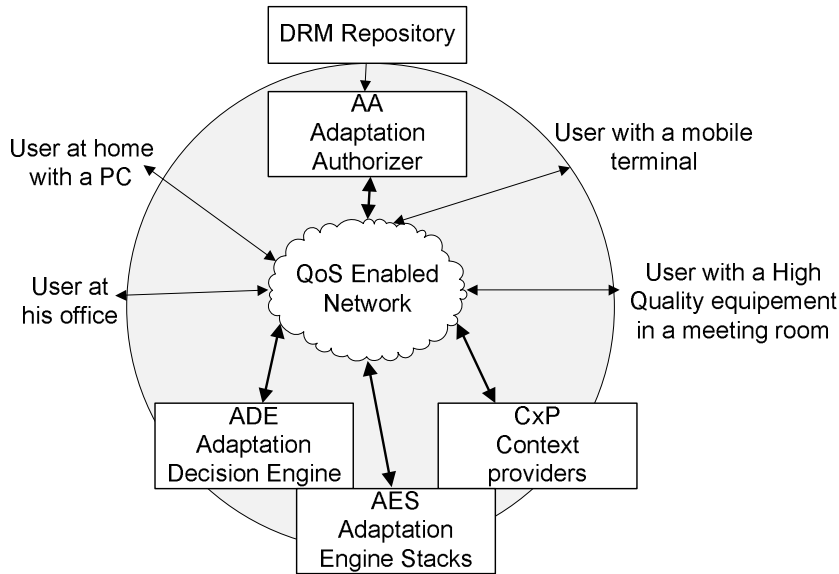


Figure 8.5: Adaptation authorization in a Virtual Collaboration Scenario

use of REL and DIA had been proposed in [158], but their focus was on syntactic integration rather than covering a complete DRM scenario.

### The Media Value Chain Ontology in the process of adaptation

The proposed solution in VISNET-II was to use the Media Value Chain Ontology to authorize the adaptation.

For this sake, the MVCO had to integrate it with a Context Aware Ontology (CAO) designed for this scenario [159]. The CAO ontology describes semantically the Usage Environment Descriptors (UED) tools. Additionally it includes the description of a media profile derived from the MPEG-7 Multimedia Description Schemes (MDSs). The main classes in CAO ontology are based on these standards, but not literally translated. The domain knowledge here conceived, can be used to model domain rules and heuristics that help to improve the quality and effectiveness of adaptation decision, also considering authorizations.

The solution consisted of the following four major modules called Adaptation Decision Engine (ADE), Adaptation Authorizer (AA), Context Providers (CxPs) and Adaptation Engine Stacks (AESs), the latter comprising a suite of Adaptation Engines (AEs) (see Fig. 8.5).

Table 8.1: Mapping of DIA/CAO/MVCO

<i>MPEG-21 DIA</i>	<i>CAO Ontology</i>	<i>MVCO</i>
UserProfile	User	User
Digital Item	Media	IPEntity
NaturalEnvironment	Natural Environment	Fact subclass
Terminal set of UEDs	Terminal	Fact subclass
Network set of UEDs	Network	Fact subclass
permittedDiaChanges		Action subclass
hasChangeConstraint		Fact subclass

CxPs are software or hardware entities that provide explicit contextual information, while ADE is the module that collects that information, inferring higher-level concepts, but it needs to interact with the authorizer (AA) to find out whether an adaptation is permitted or not. Thus, the ADE will first send an authorization request, indicating the right the user wants to execute, among other information. The AA will then issue the approval or rejection of the authorization. The AA implements a semantic engine (based on the MVCO) where the authorization decision is taken based on the application of a set of SWRL rules. These rules would check the deontic logic behind the permit: to authorize a petition, obliged propositions must hold (likewise conditions in REL), prohibited propositions must be false, and the IP value chain of authorizations will have had to be respected.

Upon the request of reproduction of a given material which has to be adapted, the created instances of the CAO ontology based on the simple XML CxPs information, together with the Media information and the desired operation and characteristics. These CAO class individuals are sent to the AA, which in turn would look up in its license store (a collection of `mvco:Permission` instances) and match for a suitable one to authorize the requested operation. Context state (terminal, network etc.) represents positive facts that can be combined in a Permit as deontic statements: requirements or prohibitions, through relations in the MVCO ontology. The matching is shown in Table 8.1, where the relation between CAO elements, the MPEG-21 DIA terms and the MVCO.

## 8.5 The Media Value Chain Ontology APIs

So far, the ontology has been presented as an OWL file. Based on RDFs and ultimately codified as an XML file, it is technologically neutral and open



for any application to use it, with no patent hurdles or obscure technologies in use. However, in order to increase and to facilitate its use, an API can also be proposed. Regardless the existence of automatic code generation tools<sup>5</sup>, this API provides the most common operations that a user of the ontology may request, and makes swifter the development of MVCO-based applications. Actually, in each of the projects mentioned in this Chapter, the MVCO API has always been present only with minor adjustments.

The API hides the details of the access interface (Jena, Pellet), making it transparent through a set of simple function calls with standard parameters. Thus, the general style of the API deals with Strings, rather than with OntClass (see Jena library) classes or similar. This is the most general form of the functions, and although the functions may receive as parameters strings not corresponding to any class or individual, validity can always be checked.

### 8.5.1 API methods

In the last version of the Media Value Chain Ontology, the following methods are defined.

```
public boolean Load(String sURI) throws IOException;
```

This method is employed to load a set of assertions from a URI. This method is additive, i.e., loading a second set of data does not replace the existing one. Data loaded includes Users, IP Entities (Digital Items), Permissions and Actions performed. The parameter is any URI making reference to an existing data repository. It can be a file, for example: `file:///test.owl`, but also database or a remote location (`http://abc.com/test.owl`).

```
public boolean Store(String sURI) throws IOException;
```

This method is employed to store a set of assertions from a URI

```
public boolean Unload();
```

This method is employed to unload the assertions in the ontology. No individuals will remain after calling this operation.

```
public boolean CreateUser(String sUserId) throws  
ExistingIdException, BadIdException;
```

---

<sup>5</sup>For example, Kazuki generates a Java API for working with OWL instances directly from an OWL ontology. See <http://projects.semwebcentral.org/projects/kazuki/>

This method is employed to create a new User, identified by a user id. The new User does not assume any role a priori. The parameter is any User identification string, provided that it starts with an alphabetic character: [a-z][A-Z], and excludes blank spaces, commas or any other non-alphanumeric character

```
public boolean DeleteUser(String sUserId) throws
    ExistingIdException, BadIdException;
```

This method is employed to unregister an existing User, identified by a user id.

```
public boolean ExecuteUser(String sActedBy, String sAction, String
    sIPEntity, String sNewIPEntity) throws BadIdException,
    IllegalActionException, ExistingIdException;
```

This method is employed for an User to execute an Action over an IPEntity, possible creating a new IPEntity.

```
public boolean CreatePermission(String sIssuedBy, String sAction,
    String sActedOver, String sActedBy) throws BadIdException,
    IllegalActionException, ExistingIdException;
```

This method is employed for an User to generate the permission to execute an Action over an IPEntity, by other User.

```
public Vector<String> Query(String sSPARQL) throws
    MalformedQueryException;
```

This method is employed to retrieve information on the ontology. The operation always retrieves information.

```
public boolean Validate(String sAction, String sActedOver,
    String sActedBy) throws BadIdException, IllegalActionException,
    ExistingIdException;
```

This method is employed to validate an action, giving a boolean answer (yes/no).

```
public Vector<String> getUsers();
```

This method is employed to retrieve the list of existing Users

```
public Vector<String> getIPEntities();
```

Operation to get the list of registered IPEntities. This method is employed to retrieve the list and nature of existing IPEntities.

```
public boolean getRightsOwner(String sAction, String sActedOver,  
    String sActedBy) throws BadIdException, ExistingIdException;
```

Operation to get the rights owner of a given IPEntity. This method retrieves the rights Owner of a given IPEntity. It does not retrieve which users have been authorised in valid Permissions. This method is redundant and given for convenience, as the query could be performed by means of the Query method.

```
public String getOrigin(String sIPEntity) throws BadIdException,  
    ExistingIdException;
```

Operation to get the IPEntity origin of a given IPEntity. This method returns the IPEntity that has given rise to the current IPEntity. If the IPEntity is a Work, a null will be returned. This method is redundant and given for convenience, as the query could be performed by means of the Query method.



**Part III**

**Conclusions**



## Chapter 9

# Conclusions

### 9.1 Outreach of this Thesis

This Thesis is the result of more than four years of research work. In this time, some of the results produced have been published in different refereed journal papers, peer reviewed conferences and workshops. Also, an important part of the work is on the way of being published as an International Standard. All of it is listed in the next section.

In particular, the Media Value Chain Ontology described in Chapter 6 has been proposed to constitute Part 19 of the standard MPEG-21. Having started the process in July 2007, it has successfully passed the stages of Working Draft, First Committee Draft and Final Committee Draft, being currently in the balloting for becoming Final Draft International Standard.

A first version of this ontology and its derived API had been already approved by the Digital Media Project, and included as a part of the Interoperable DRM Platform specification.

Derived from the Media Value Chain Ontology, the concept of a service provided on top of the ontology integrated as an element in the DRM architecture, is a new concept contributed by the work presented in this Thesis. The shadowed boxes in Fig. 8.3 or Fig. 5.8 are the implementation of the ideas of this Thesis, and they have been approved and included in the architectures of different projects.

Other minor contributions have been reflected in other parts of the MPEG standards, like in the MPEG-M (the MXM). There, mechanisms for integrating the MVCO within the rest of the MPEG-21 framework in practical applications were specified, as well as some nuances in the license authorization mechanism to allow future different authorization mechanisms

like the Semantic Authorizer.

The author of this Thesis wishes that these contributions constitute a step more towards the fairness of digital media exchange in the Digital Era.

## 9.2 Contributions

### 9.2.1 Refereed publications

Listed in chronological order of publication.

- Victor Rodríguez, Marc Gauvin and Jaime Delgado, An Ontology for the Expression of Intellectual Property Entities and Relations, Proceedings of the 5th International Workshop on Security in Information Systems (WOSIS 2007), Funchal, April 2007
- Victor Rodríguez, Jaime Delgado and Eva Rodríguez, From Narrative Contracts to Electronic Licenses: A Guided Translation Process for the Case of Audiovisual Content Management, in Proc. of the 3rd Int. Conference on Automated Production of Cross Media Content for Multi-Channel Distribution (AXMEDIS 2007), Barcelona, November 2007
- Victor Rodríguez and Jaime Delgado, Multimedia Content Distribution Governed by Ontology-Represented Contracts, Workshop on Multimedia Ontologies and Artificial Intelligence Techniques in Law, (MOAIL 2007) Leiden, December 2007
- Victor Rodriguez, Anna Carreras, Vitor Barbosa, Jaime Delgado and Maria Teresa Andrade, A Semantic Model for the Authorisation of Context-Aware Content Adaptation, in Proc. of the 3rd Int. Conference on Semantic and Digital Media Technologies (SAMT 2008), Koblenz, December 2008
- Eva Rodríguez, Victor Rodríguez, Anna Carreras and Jaime Delgado, A Digital Rights Management approach to privacy in online social networks, Workshop on on Privacy and Protection in Web-based Social Networks (within ICAIL2009), Barcelona, June 2009
- Víctor Rodríguez, Anna Carreras, Eva Rodríguez and Jaime Delgado, Applications to Improve Privacy on Online Social Networks, Proc. of the First Workshop on Law and Web 2.0, Antoni Roig (ed.), ISSN 20135017 September 2009



- Víctor Rodríguez and Jaime Delgado, Semantic Expression and Execution of B2B Contracts on Multimedia Content, in Proc. of the 4th Int. Conference on Semantic and Digital Media Technologies (SAMT 2009), Graz, December 2009
- Victor Rodríguez and Jaime Delgado, A Media Value Chain Ontology for MPEG-21, IEEE Multimedia, vol. 16, n°4, pp. 44-51, ISSN: 1070-986X, October-December 2009.

### 9.2.2 Standardization contributions

#### In the MPEG group

- M. Gauvin, J. Delgado, E. Rodríguez and V. Rodríguez, Requirements for a machine readable ontology, Technical Report, Input Document M14707 for the ISO/IEC JTC1/SC 29/WG11, Moving Pictures Experts Group, July 2007
- M. Gauvin, J. Delgado, V. Rodríguez, A Common Core IP Model, ISO/IEC JTC 1/SC 29/WG 11 MPEG 83, January 2008
- J. Delgado, E. Rodríguez, V. Rodríguez-Doncel, M. Baldinato, F. Bixio , Use of MPEG-21 REL for the expression of Audiovisual Contracts, ISO/IEC JTC 1/SC 29/WG 11 MPEG 84, April 2008
- J. Delgado, E. Rodríguez, V. Rodríguez-Doncel, How to express Exclusivity in MPEG-21 REL, ISO/IEC JTC 1/SC 29/WG 11 M15431, MPEG 84, April 2008
- M. Gauvin, J. Delgado, V. Rodriguez, Ontology for IP Media Networks: A response to the MVCO CfP, ISO/IEC JTC 1/SC 29/W11 M15631, July 2008
- W. Allasia, M. T. Andrade, M. Bober, H. Castro, J. Chen, F. Chiariglione, M. Cosenza, J. Delgado, A. Difino, V. Rodríguez, F. Gallo, M. Gauvin, T. Huang, J. Lee, Y. K. Lim, D. Oliver Kropp, M. Mattavelli, M. Milanese, J. Naito, M. Preda, G. Ruffo, R. Schifanella, M. Springer, C. Timmerer, X. Wang, Response to MXM CfP, ISO/IEC JTC 1/SC 29/WG 11 MPEG, July 2008
- W. Allasia, M. T. Andrade, M. Bober, H. Castro, J. Chen, F. Chiariglione, M. Cosenza, J. Delgado, A. Difino, V. Rodríguez, F. Gallo,

- M. Gauvin, T. Huang, J. Lee, Y. K. Lim, D. Oliver Kropp, M. Mat-tavelli, M. Milanesio, J. Naito, M. Preda, G. Ruffo, R. Schifanella, M. Springer, C. Timmerer, X. Wang, Collection of MXM Use Cases , ISO/IEC JTC 1/SC 29/WG 11 MPEG, July 2008
- W. Allasia, M. T. Andrade, M. Bober, H. Castro, J. Chen, F. Chia-riglione, M. Cosenza, J. Delgado, A. Difino, V. Rodríguez, F. Gallo, M. Gauvin, T. Huang, J. Lee, Y. K. Lim, D. Oliver Kropp, M. Mat-tavelli, M. Milanesio, J. Naito, M. Preda, G. Ruffo, R. Schifanella, M. Springer, C. Timmerer, X. Wang, Proposal of MXM Technologies , ISO/IEC JTC 1/SC 29/WG 11 MPEG, July 2008
  - W. Allasia, M. T. Andrade, M. Bober, H. Castro, J. Chen, F. Chia-riglione, M. Cosenza, J. Delgado, A. Difino, V. Rodríguez, F. Gallo, M. Gauvin, T. Huang, J. Lee, Y. K. Lim, D. Oliver Kropp, M. Mat-tavelli, M. Milanesio, J. Naito, M. Preda, G. Ruffo, R. Schifanella, M. Springer, C. Timmerer, X. Wang, Proposal of MXM Technologies (M15689), ISO/IEC JTC 1/SC 29/WG 11 MPEG, July 2008
  - M. Gauvin, J. Delgado, V. Rodríguez Doncel, M. Choi, Media Value Chain Ontology (Working Draft), ISO/IEC JTC 1/SC 29/WG 11 MPEG 85, July 2008
  - M. Gauvin, J. Delgado, V. Rodríguez Doncel, M. Choi, Media Value Chain Ontology (Committee Draft), ISO/IEC JTC 1/SC 29/WG 11 MPEG 86, October 2008
  - M. Gauvin, J. Delgado, V. Rodríguez Doncel, M. Choi, ISO/IEC CD 21000-19 Media Value Chain Ontology, ISO/IEC JTC 1/SC 29/WG 11 MPEG 86, October 2008
  - J. Delgado, E. Rodríguez, V. Rodríguez-Doncel, S. Llorente, R. Barrio V. Torres, DMAG-UPC Comments on WD2.0 of MXM API, ISO/IEC JTC 1/SC 29/WG 11 MPEG, Lausanne, Switzerland, February 2009
  - V. Rodríguez-Doncel, J. Delgado, R. Tous, Presentation of the W3C Media Annotations Working Group, ISO/IEC JTC 1/SC 29/WG 11 MPEG 87, February 2009
  - M. Gauvin, J. Delgado, V. Rodríguez Doncel, M. Choi, Draft Text of ISO/IEC FCD 21000-19 Media Value Chain Ontology, ISO/IEC JTC 1/SC 29/WG 11 MPEG 87, February 2009

### Contributions in the DMP group

These two contributions are the most relevant.

- M. Gauvin, J. Delgado and V. Rodriguez, Proposed RRD Text for Approved Document No 2 - Technical Reference: Architecture, v. 2.1, Digital Media Project DMP0952/AHG40, July 2007
- M. Gauvin, J. Delgado, V. Rodriguez, Proposed RRD Text for Approved Document No 3 - Technical Specification: Interoperable DRM Platform, ver. 2.1, Digital Media Project 0953/AHG41, March 2007

### 9.3 Future lines of work

This Thesis constitutes a closed work, but there is some reasearch to be done to enhance its applicability and there is some exploration to be done to check whether the same ideas can be extrapolated to topics like privacy management, document management systems and others. Also, the efficiency in the transactions has not been dealt in this work, while it should be a critical factor at the exploitation time.

First, the ontology-based applications should be assured to run swiftly regardless the number of exemplars. Efficiency is (one of) the Achilles heel of the Semantic Web, and processing massive data with non-optimized software, as frequently ontology software comes, compromises the whole architecture. In the last years, efficient triple stores (RDF databases) have appeared, and improved implementations of the ontology reasoners, both commercial and open source. The whole architecture remains, nevertheless, stuck to Java platforms and to a short variety of deployed applications, and there is a chicken-egg problem regarding ontology use and its widespread use. Efficient and scalable ad-hoc implementations for the media value chain ontology handling could be then sketched and implemented, and although they outreached the scope of this Thesis, this topic should be dealt as soon as a commercial exploitation of MVCO-based applications are envisaged.

Second, the semantic authoriser should be fully implemented with a complete set of clauses and events able to be handled. For this regard, interpretation of many elements could be referred to those in external ontologies, like the DBpedia (the Wikipedia written as RDF triples), WordNet, the Event Ontology etc. This implementation would lead to the ability of enforcing many clauses that might be absurd (this track can only be played on days of full moon, or lands which have not had a war in the last twenty years,

or in rainy days etc.) but whose use can never be suspected (for example, some tracks of pornographic or political content can be banned in certain countries etc.)

And finally, the ontology could be extrapolated to other areas. Privacy is a concern of increasing importance, as each time in the Internet era personal data is given in more and more sites. The costumer, an Internet user, makes use of email services, social networks, internet shopping and many other places where personal information is given, sometimes of high value and sometimes extremely sensitive, like personal data. For the case of social network, the social network provider has the responsibility of keeping the privacy preserved, but for the moment there is not a clear model stating to what extent and for how long. We discern here a similar case to the intellectual property one, handled in this Thesis. The Internet user can be considered a creator of his personal data, and he is his only rights owner, but he is willing to transfer it according to a certain terms of use (i.e. not transferring the data to third parties). For the moment these terms of use are only narrative text, and different for each network provider, but a value chain of personal information could be thus traced and eventually handled like the media value chain. An introductory work in this like has already been presented [160], but this new, fast evolving field has to be further studied.

## **Acronyms**

**API** Application Programming Interface

**CEL** Contract Expression Language

**DIG** DL Implementation Group

**DL** Description Logics

**DMCA** Digital Millennium Copyright Act

**DRM** Digital Rights Management

**EDI** Electronic Data Interchange

**FOAF** Friend Of A Friend

**FRBR** Functional Requirements for Bibliographic Records

**IP** Intellectual Property

**IPMP** Intellectual Property Management and Protection

**MPEG** Moving Pictures Experts Group

**OASIS** Organization for the Advancement of Structured Information Standards

**OWL** Ontology Web Language

**RDD** Rights Data Dictionary

**RDF** Resource Description Format

**RDFS** Resource Description Framework Schema

**REL** Rights Expression Language

**RFC** Request for Comments

**SKOS** Simple Knowledge Organisation Systems

**SOAP** Simple Object Access Protocol

**SPARQL** SPARQL Protocol And Query Language

**URI** Uniform Resource Identifier

**URL** Uniform Resource Locator

**WIPO** World Intellectual Property Organization

**XML** eXtensible Markup Language

**W3C** World Wide Web Consortium

# Bibliography

- [1] Lawrence Lessig. *Free Culture: The Nature and Future of Creativity*. Ed. The Penguin Press, New York 2004. ISBN 978-0143034650.
- [2] Walter Benjamin. *The work of art in the age of reproducibility*. Ed. Penguin Books, 2008. ISBN 978-0141036199.
- [3] Natali Helberger, Francisco Javier Cabrera, and Susanne Nikoltchev. Copyright and related rights in the audiovisual sector. Technical report, IRIS. Institute for Information Law. European Audiovisual Observatory, 2000.
- [4] World Intellectual Property Organization. Understanding Copyright and Related Rights. Technical Report 909e, World Intellectual Property Organization, 2008. ISBN 92-805-1265-4.
- [5] World Intellectual Property Organization. Intellectual Property Handbook: Policy, Law and Use. Technical Report 489e, World Intellectual Property Organization, 2004. ISBN 92-805-1291-7.
- [6] The United States Digital Millennium Copyright Act (DCMA), in. <http://www.loc.gov/copyright/legislation/hr2281.pdf>, 8 October 1998.
- [7] Richard Stim. *Patent, copyright and trademark*. Ed. Nolo Press 8th Edition, Berkeley 2006. ISBN 1-4133-0196-7.
- [8] Dieter Fensel. *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. Springer, 1st edition, August 2001. ISBN 978-3540416029.
- [9] Sergei Nirenburg and Victor Raskin. *Ontological Semantics*. MIT Press, 2004. ISBN 978-0-262-14086-7.

- [10] John Woods and Dov Gabbay. *Handbook of the History of Logic*. Ed. Elsevier, 2004. ISBN 0-444-51596-8.
- [11] Alfredo Deaño. *Introducción a la lógica formal*. Ed. Alianza Universidad, 1974. ISBN 84-206-8011-7.
- [12] Alonzo Church. An unsolvable problem of elementary number theory. *J. Symbolic Logic*, 1:73–74, 1936.
- [13] Raymond M. Smullyan. *First-Order Logic*. Ed. Dover Publications, 1968. ISBN: 04-866-8370-2.
- [14] ISO/IEC JTC 1/SC 32. ISO/IEC 24707:2007, Information technology — Common Logic (CL): A framework for a family of logic-based languages, 2007.
- [15] George Edward Hughes. *An introduction to modal logics*. Ed. Routledge, 1990. ISBN 978-0415043137.
- [16] Saul A. Kripke. A completeness theorem in modal logic. *J. Symbolic Logic*, 24(1):1–14, March 1959.
- [17] Pablo F. Castro and Tom S. E. Maibaum. A complete and compact propositional deontic logic. In *Theoretical Aspects of Computing*, volume 4711/2007, pages 109–123. Springer Berlin / Heidelberg, 2007. ISBN 978-3-540-75290-5.
- [18] Jesús Rodríguez. *Lógica deóntica: Concepto y Sistemas*. Ed. Secretariado de Publicaciones Universidad de Valencia, 1978. ISBN: 84-370-0057-2.
- [19] Jörg Hansen, Gabriella Pigozzi, and Leendert van der Torre. Ten philosophical problems in deontic logic. In Guido Boella, Leon van der Torre, and Harko Verhagen, editors, *Normative Multi-agent Systems*, number 07122 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2007.
- [20] Carlo Biagioli, Giovanni Sartor, and Daniela Tiscornia (eds.). KED: A deontic theorem prover. In *Proc. of the Workshop on Legal Application of Logic Programming*, pages 60–76. IDG, 1994.
- [21] Guido Governatori and Zoran Milosevic. A formal analysis of a business contract language. *Int. J. Cooperative Inf. Syst.*, 15(4):659–685, 2006.



- [22] Alan S. Abrahams, David M. Eyers, and Jean Bacon. Mapping legal cases to RDF named graphs using a minimal deontic ontology for computer-assisted legal inference. In *Proc. of the First Int. Conf. on AI and Law (ICAIL07), Workshop on Semantic Web technology for Law*, pages 11–20, Stanford, Palo Alto, CA, June 2007.
- [23] Franz Baader Deborah L. McGuinness Daniele Nardi Peter F. Patel-Schneider . *The Description Logic Handbook: Theory Implementation and Applications*. Cambridge University Press, 2003. ISBN 978-0521781763.
- [24] A. Collins and M. Quillian. Retrieval time from semantic memory. *Journal of Verbal Learning and Verbal Behavior*, 8(2):240–247, April 1969.
- [25] U. Hustadt, R. A. Schmidt, and L. Georgieva. A survey of decidable first-order fragments and description logics. *Journal of Relational Methods in Computer Science*, 1, 2008.
- [26] Tim Berners-Lee. *Weaving the Web*. Ed. Harper, San Francisco 1999. ISBN 0062515861.
- [27] Kevin T. Smith Daconta, Michael C. and Leo J. Obrst. *The Semantic Web: A Guide to the Future of XML, Web Services and Knowledge Management*. John Wiley and Sons Inc., 2003. ISBN: 0471432571.
- [28] IETF Request For Comments 3986. RFC 3986, Uniform Resource Identifier (URI): Generic Syntax. Technical report, IETF, January 2005.
- [29] Eran Hammer and Will Norris. Extensible Resource Descriptor (XRD) Version 1.0 Committee Draft 01. Technical report, OASIS, October 2009. <http://docs.oasis-open.org/xri/xrd/v1.0/cd01/xrd-1.0-cd01.html>.
- [30] R. Rodgers. The ARK Identifier Scheme. Technical report, California Digital Library, 2008. <http://www.cdlib.org/inside/diglib/ark/arkspec.html>.
- [31] ISO/IEC JTC 1/SC 29. ISO/IEC 21000-3:2004, Information technology — Multimedia framework (MPEG 21) — Part 3: Digital Item Identification, 2004.

- [32] T. Bray, J. Paoli, and C. Sperberg-McQueen. Extensible Markup Language (XML) 1.0. W3C Recommendation, World Wide Web Consortium (W3C), 1998. <http://www.w3.org/TR/1998/REC-xml-19980210>.
- [33] James Clark and Murata Makoto. RELAX NG Specification. Committee specification, OASIS, December 2001. <http://relaxng.org/spec-20011203.html>.
- [34] Henry S. Thompson, David Beech, Murray Maloney, and Noah Mendelsohn. XML Schema Part 1: Structures Second Edition. W3C Recommendation, W3C, October 2004. <http://www.w3.org/TR/xmlschema-1/>.
- [35] Paul V. Biron and Ashok Malhotra. XML schema part 2: Datatypes second edition, October 2004. <http://www.w3.org/TR/xmlschema-2/>.
- [36] Andrew Layman Tim Bray, Dave Hollander and Richard Tobin. Namespaces in XML 1.0 (Second Edition). W3C Recommendation, W3C, August 2006.
- [37] Frank Manola and Eric Miller. RDF Primer. W3C Recommendation, W3C, February 2004. <http://www.w3.org/TR/rdf-primer/>.
- [38] Shelley Powers. *Practical RDF*. Ed. O'Reilly, 2003. ISBN: 0-596-00263-7.
- [39] David Beckett and Tim Berners Lee. Turtle - Terse RDF Triple Language. W3C Team Submission, W3C, January 2008. <http://www.w3.org/TeamSubmission/turtle/>.
- [40] David Beckett. RDF/XML Syntax Specification (Revised). W3C Recommendation, W3C, February 2004. <http://www.w3.org/TR/2004/REC-rdf-syntax-grammar/>.
- [41] Dan Brickley and Ramanathan Guha. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, W3C, February 2004. <http://www.w3.org/TR/rdf-schema>.
- [42] Scott Boag et al. XQuery 1.0: An XML Query Language W3C Recommendation. Technical report, W3C, January 2007. <http://www.w3.org/TR/xquery/>.

- [43] E. Prud'hommeaux and A. Seaborne (eds.). SPARQL, SPARQL Protocol and RDF Query Language. W3C Working Draft, W3C, October 2006. <http://www.w3.org/TR/rdf-sparql-query/>.
- [44] Andy Seaborne, Geetha Manjunath. SPARQL/Update: A language for updating RDF graphs. Technical report, HP, July 2007.
- [45] Guus Schreiber. (eds.) M. Dean. OWL Web Ontology Language Reference. W3C Recommendation, W3C, 2004. <http://www.w3.org/TR/owl-ref/>.
- [46] Peter F. Patel-Schneider, Patrick Hayes, and Ian Horrocks (eds.). OWL Web Ontology Language Semantics and Abstract Syntax. W3C Recommendation, W3C, February 2004. <http://www.w3.org/TR/owl-semantics/>.
- [47] W3C OWL Working Group. OWL 2 Web Ontology Language Document Overview. W3C Recommendation, W3C, October 2009. <http://www.w3.org/TR/owl-overview/>.
- [48] Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Groszof, and Mike Dean. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Member Submission, W3C, 2004.
- [49] Alistair Miles and Sean Bechhofer. SKOS Simple Knowledge Organization System Reference. W3C Recommendation, W3C, August 2009. <http://www.w3.org/TR/skos-reference/>.
- [50] Akrivi Katifori, Constantin Halatsis, George Lepouras, Costas Vassilakis, and Eugenia Giannopoulou. Ontology visualization methods—a survey. *ACM Comput. Surv.*, 39(4):10, 2007.
- [51] Jan Wielemaker, Guus Schreiber, and Bob Wielinga. Prolog-based infrastructure for rdf: Scalability and performance. In *In: The Semantic Web - ISWC 2003*, pages 644–658. Springer-Verlag, 2003.
- [52] Samuil Angelov and Paul Grefen. The business case for B2B e-contracting. In *Proc. of the 6th Int. Conf. on Electronic Commerce (ICEC '04)*, pages 31–40, New York, NY, USA, 2004. ACM.
- [53] Melvin Aron Eisenberg. The Emergence of Dynamic Contract Law. *Theoretical Inquiries in Law*, 2(1), January 2001.

- [54] Samuil Angelov and Paul Grefen. An e-contracting reference architecture. *J. Syst. Softw.*, 81(11):1816–1844, 2008.
- [55] Frank Griffel, Marko Boger, Harald Weinreich, Winfried Lamersdorf, and Michael Merz. Electronic Contracting with COSMOS - How to Establish, Negotiate and Execute Electronic Contracts on the Internet. In Z. Milosevic C. Kobryn, C. Atkinson, editor, *2nd Int. Enterprise Distributed Object Computing Workshop (EDOC98)*, page 10, November 1998.
- [56] Yao Hua Tan and Thoen W. DocLog: an electronic contract representation language. In *Proc. of the 11th Int. Workshop on Database and Expert Systems Applications*, September 2000.
- [57] Content Reference Forum. CEL: Contract Expression Language. (rev. 1). <http://www.crforum.org/candidate/CELWG002-celspec.doc>.
- [58] Vandana Kabilan and Paul Johannesson. Semantic representation of contract knowledge using multi-tier contract ontology. In *Proc. of Semantic Web and Databases workshop, (SWDB 2003)*, 2003.
- [59] Silvia Llorente, Jaime Delgado, Eva Rodríguez, Rubén Barrio, Isabella Longo, and Franco Bixio. Generation of standardised rights expressions from contracts: An ontology approach? In *OTM Workshops*, 2005. LNCS 3762.
- [60] Yalan Yan, Jinlong Zhang, and Mi Yan. Ontology Modeling for Contract: Using OWL to Express Semantic Relations. In *Proc. of the 10th IEEE Int. Enterprise Distributed Object Computing Conference (EDOC06)*, pages 409–412, 2006. ISSN 0-7695-2558-X.
- [61] Adrian Paschke, Martin Bichler, and Jens Dietrich. ContractLog: An Approach to Rule Based Monitoring and Execution of Service Level Agreements. In *RuleML*, pages 209–217, 2005.
- [62] Michal Morciniec, Michal Morciniec, Mathias Salle, Mathias Salle, Brian Monahan, and Brian Monahan. Towards regulating electronic communities with contracts. In *Proc. of the 2nd Workshop on Norms and Institution in Multiagent Systems*, May 2001.
- [63] P.R. Krishna, K. Karlapaplem, and A.R. Dani. *From Contracts to E-Contracts: Modeling and Enactment*, chapter 6, pages 386–387. Springer Science, October 2005.

- [64] Benjamin N. Grosf and Terrence C. Poon. Sweetdeal: representing agent contracts with exceptions using xml rules, ontologies, and process descriptions. In *WWW '03: Proc. of the 12th Int. Conf. on World Wide Web*, pages 340–349, New York, NY, USA, 2003. ACM.
- [65] S. Neal, J. Cole, P. F. Lington, Z. Milosevic, S. Gibson, and S. Kulka-rni. Identifying requirements for business contract language: a monitoring perspective. *Enterprise Distributed Object Computing Conf., IEEE International*, 0:50, 2003.
- [66] Lawrence A. Cunningham. Language, Deals and Standards: The Future of XML Contracts. *Washington University Law Review*, 2005. SSRN 900616.
- [67] Birgit Hofreiter, Christian Huemer, and Klaus dieter Naujok. UN/CE-FACT's Business Collaboration Framework - Motivation and Basic Concepts. In *Proc. of MKWI'04 Track on Co-ordination in Value Creation networks / Agent Technology for Business Applications, LNI GITO (2004)*, 2004.
- [68] Cristian Prisacariu and Gerardo Schneider. A formal language for electronic contracts. In Marcello Bonsangue and Einar Broch Johnsen, editors, *9th IFIP Int. Conf. on Formal Methods for Open Object-Based Distributed Systems (FMOODS'07)*, volume 4468 of *Lecture Notes in Computer Science*, pages 174–189. Springer, June 2007.
- [69] Peter Meyer. A Proposed XML Standard for Contract Documents. In *Proc. of the Syntax to Semantics XML 2005 Conf.*, 2005.
- [70] Peter Meyer Laurence Leff. OASIS eContracts Version 1.0. committee specification. Technical report, OASIS, April 2007.
- [71] Joeri M. Mol and Nachoem M. Wijnberg. Value chain envy: explaining new entry and vertical integration in popular music. Research Report 02B62, University of Groningen, Research Institute SOM, 2002.
- [72] Antonio Hidalgo, José Albors, and Vicente Lopez. Design and development challenges for an e2e drm content business integration platform. *International Journal of Information Management*, 29(5):389 – 396, 2009.
- [73] Cornelia C. Krueger and Paula M. C. Swatman. Who are the internet content providers? In *Digital Communities in a Networked Society*:

- eCommerce, eBusiness, and eGovernment, The Third IFIP Conf. on E-Commerce, E-Business, E-Government (I3E 2003)*, pages 27–38, 2003.
- [74] Arto Toumola. Disintermediation and reintermediation of the sound recording value chain: Two case studies. *Journal of Media Business Studies*, 1(1):27–46, 2004.
- [75] K. Alves and K. Michael. The Rise and Fall of Digital Music Distribution Services: a Cross-Case Comparison of MP3.com, Napster and Kazaa. In Ed. Narciso Cerpa, editor, *Proc. of the Collaborative Electronic Commerce Technology and Research Conf.*, 2005.
- [76] Stephan Baumann and Melanie Dulong de Rosnay. Music life cycle support through ontologies. In Valerie Barr and Zdravko Markov, editors, *Proc. of the Seventeenth Int. Florida Artificial Intelligence Research Symposium Conf. (FLAIRS)*, Miami Beach, Florida, USA, 2004. AAAI Press.
- [77] Roberto García and Rosa Gil. Content value chains modelling using a copyright ontology. *Information Systems*, In Press, Corrected Proof:–, 2009.
- [78] Junseok Lee, Seong Oun Hwang, Sang-Won Jeong, Ki Song Yoon, Chang Soon Park, and Jae-Cheol Ryou. A DRM Framework for Distributing Digital Contents through the Internet. *ETRI Journal*, 25(6):423–436, December 2003.
- [79] J. Lee, J. Park, K. Yoon, and J. Ryou. Metadata in Contract System for Protecting Rights of Distributions Subjects. In Acta Press, editor, *Internet and Multimedia Systems and Applications*, 2003.
- [80] Yves Raimond, Samer Abdallah, Mark Sandler, and Frederick Giasson. The music ontology. In *Proc. of the 8th Int. Conf. on Music Information Retrieval (ISMIR 2007)*, Vienna, Austria, September 2007.
- [81] Yves Raimond, Christopher Sutton, and Mark Sandler. Interlinking music-related data on the web. *IEEE MultiMedia*, 16(2):52–63, 2009.
- [82] Nick Crofts, Martin Doerr, Tony Gill, Stephen Stead, Matthew Stiff. Definition of the CIDOC Conceptual Reference Model: version 5.1. Technical report, ICOM/CIDOC Documentation Standards Group. CIDOC CRM Special Interest Group, March 2009.

- [83] Martin Oischinger Guenther Goerz, Bernhard Schiemann. An Implementation of the CIDOC Conceptual Reference Model in OWL-DL. In *Proc. of the Annual Conf. of CIDOC*, 2008.
- [84] Renato Iannella. *Rights Management and Licensing Multimedia Services*, volume 120, pages 327–349. Springer, June 2008.
- [85] Gwenaël Doërr and Ton Kalker. Design rules for interoperable domains: controlling content dilution and content sharing. In *Proc. of the 8th ACM workshop on Digital rights management (DRM '08)*, pages 39–50, New York, NY, USA, 2008. ACM.
- [86] Open Mobile Alliance. DRM Specification Version 2.1. Technical report, Open Mobile Alliance, August 2008.
- [87] Marvin L. Smith. Digital rights management & protecting the digital media value chain. In *MUM '04: Proc. of the 3rd Int. Conf. on Mobile and ubiquitous multimedia*, pages 187–191, New York, NY, USA, 2004. ACM.
- [88] Rights and Rules Working Group. OeBF Rights Grammar Requirements. Required Rights Features for Digital Books. Technical report, Open eBook Forum, March 2003.
- [89] Carlos Serrão, Daniel Neves, Trevor Barker, Massimo Balestri, and Panos Kudumakis. Open sDRM. An open and secure Digital Rights Management solution. In *Proc. of the IADIS Int. Conf. on e-Society*, June 2003.
- [90] Víctor Torres, Carlos Serrão, Miguel Sales Dias, and Jaime Delgado. Open DRM and the Future of Media. *IEEE MultiMedia*, 15(2):28–36, 2008.
- [91] Sun Microsystems. DReaM-CAS Client Specification Version 1.0. Technical report, Open Media Commons, January 2007.
- [92] R.H. Könen, J. Lacy, M. Mackay, and S. Mitchell. The long march to interoperable digital rights management. *Proceedings of the IEEE*, 92(6):883–897, June 2004.
- [93] Víctor Torres, Eva Rodríguez, and Jaime Delgado. Reporting events in a multimedia content distribution and consumption system. In *Distributed Media Systems*, pages 44–50, 2008.

- [94] Eduard Turcan, Nahid Shahmehri, and Germano Caronni. Usability and security for drm architectures. In *Proc. of 5th e-Smart Conf. and Demos*. Strategiestm, 2004.
- [95] Jaime Delgado, Víctor Torres, Silvia Llorente, and Eva Rodríguez. Rights and trust in multimedia information management. In *Communications and Multimedia Security*, pages 55–64, 2005.
- [96] Nicholas Paul Sheppard. On implementing mpeg-21 intellectual property management and protection. In *Proc. of the 2007 ACM workshop on Digital Rights Management (DRM '07)*, pages 10–22, New York, NY, USA, 2007. ACM.
- [97] ISO/IEC JTC 1/SC 29. ISO/IEC 21000-5:2004, Information technology — Multimedia framework (MPEG 21) — Part 5: Rights Expression Language, 2004.
- [98] ISO/IEC JTC 1/SC 29. ISO/IEC 21000-6:2004, Information technology — Multimedia framework (MPEG-21) — Part 6: Rights Data Dictionary, 2004.
- [99] MPEG Requirements Group. N10237 mpeg extensible middleware vision. Technical report, ISO/IEC JTC 1/SC 29/WG 11, 2008.
- [100] The Digital Media Project. The Digital Media Manifesto. In: <http://manifesto.chiariglione.org/>, September 2003.
- [101] The Digital Media Project. Approved Document No. 1, Technical Reference: Value Chain Functions and Requirements, Version 3.2. <http://www.dmpf.org/open/dmp1201.zip>, October 2007.
- [102] The Digital Media Project. Approved Document No. 2, Technical Reference: Architecture, Version 3.2. <http://www.dmpf.org/open/dmp1202.zip>, October 2007.
- [103] The Digital Media Project. Approved Document No. 3, Technical Specification: Interoperable DRM Platform, Version 3.2. <http://www.dmpf.org/open/dmp1203.zip>, October 2007.
- [104] The Digital Media Project. Approved Document No. 4, Technical Specification: Use Cases and Value Chains, Version 3.2. <http://www.dmpf.org/open/dmp1204.zip>, October 2007.



- [105] The Digital Media Project. Approved Document No. 5, Technical Specification: Certification and Registration Authorities, Version 3.2. <http://www.dmpf.org/open/dmp1205.zip>, October 2007.
- [106] The Digital Media Project. Approved Document No. 6, Technical Reference: Terminology, Version 3.2. <http://www.dmpf.org/open/dmp1206.zip>, October 2007.
- [107] The Digital Media Project. Approved Document No. 7, Technical Specification: Reference software, Version 3.2. <http://www.dmpf.org/open/dmp1207.zip>, October 2007.
- [108] Xiaofan Chen and Tiejun Huang. Interoperability issues in drm and dmp solutions. In *Multimedia and Expo, 2007 IEEE International Conf. on*, pages 907–910, July 2007.
- [109] ISO/IEC 23006-1, Information technology. MPEG-M (MPEG Extensible middleware) - Part 1: MXM Architecture and Technologies, Committee Draft. <http://mxm.wg11.sc29.org>, March 2009.
- [110] ISO/IEC 23006-2, Information technology. Information technology - MPEG-M (MPEG Extensible middleware) - Part 2: MXM API, Committee Draft. <http://mxm.wg11.sc29.org>, March 2009.
- [111] ISO/IEC 23006-3, Information technology. MPEG-M (MPEG Extensible Middleware) - Part 3: MXM Conformance and Reference Software. <http://mxm.wg11.sc29.org>, March 2009.
- [112] ISO/IEC 23006-4, Information technology. MPEG-M (MPEG Extensible Middleware) - Part 4: MXM Protocols. <http://mxm.wg11.sc29.org>, March 2009.
- [113] ISO/IEC 29116-1, Information technology - Supplemental Media Technologies - Part 1: MPEG Extensible middleware (MXM) Protocols, Committee Draft, March 2009.
- [114] ISO/IEC 15938-12, Information Technology - Multimedia Content Description Interface - Part 12: Query Format, 2004.
- [115] J. Thomas-Kerr, I. Burnett, and C. Ritz. Format-independent rich media delivery using the bitstream binding language. *Multimedia, IEEE Transactions on*, 10(3):514–522, April 2008.

- [116] Andrew Tokmakoff, François-Xavier Nuttall, and Kyunghhee Ji. Mpeg-21 event reporting: Enabling multimedia e-commerce. *IEEE Multi-Media*, 12(4):50–59, 2005.
- [117] Mark J. Stefik. Patent issued to Xerox Corporation. System for controlling the distribution and use of digital work having attached usage rights where the usage rights are defined by a usage rights grammar. US Patent 5,715,403, 1998.
- [118] Xin Wang, Guillermo Lao, Thomas DeMartini, Hari Reddy, Mai Nguyen, and Edgar Valenzuela. XRML - extensible rights markup language. In *Proc. of the 2002 ACM workshop on XML security (XMLSEC '02)*, page 71–79. ACM Press. 69, 2002.
- [119] ISO/IEC JTC 1/SC 29. ISO/IEC 21000-5:2004/Amd 1:2007, MAM (Mobile And optical Media) profile, 2007.
- [120] ISO/IEC JTC 1/SC 29. ISO/IEC 21000-5:2004/Amd 2:2007, DAC (Dissemination And Capture) profile, 2007.
- [121] ISO/IEC JTC 1/SC 29. ISO/IEC 21000-5:2004/Amd 3:2008, OAC (Open Access Content) profile, 2007.
- [122] Renato Ianella (ed.). Open Digital Rights Language v.1.1. Technical report, W3C, 19 September 2002.
- [123] Morgan V. Cundiff. An introduction to the Metadata Encoding and Transmission Standard (METS). *Library Hi Tech*, 22:52 – 64, 2004.
- [124] ETSI TS 102 822-5 Broadcast and On line Services. Search, select, and rightful use of content on personal storage systems. Part 5: Rights Management and Protection (RMP) Information for Broadcast Applications. Technical report, European Telecommunications Standards Institute, 2005.
- [125] Hal Abelson, Ben Adida, Mike Linksvayer, and Nathan Yergler. ccREL: The Creative Commons Rights Expression Language, March 2008.
- [126] Cheun Ngen Chong, Ricardo Corin, Sandro Etalle, Pieter Hartel, Willem Jonker, and Yee Wei Law. Licensescript: A novel digital rights language and its semantics. *Web Delivering of Music, International Conference on*, 0:122, 2003.

- [127] Huiying Li Yuzhong Qu, Xiang Zhang. An ontology-based rights expression language. In *Proc. of the 13th international World Wide Web.*, 2004.
- [128] Jaime Delgado, Isabel Gallego, and Roberto García. Use of semantic tools for a digital rights dictionary. In *EC-Web*, pages 338–347, 2004.
- [129] Adam Pease and Godfrey Rust. Formal Ontology for Media Rights Transactions. Technical report, Rightscom, 2007.
- [130] A. Michael Noll. The Digital Computer as a Creative Medium. *IEEE Spectrum*, 4(10):89–95, October 1967.
- [131] Mariano Fernández. Overview of the methodologies for building ontologies. In *Proc. of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods (KRR5)*, August 1999.
- [132] D. Jones, T. Bench-Capon, and P. Visser. Methodologies for ontology development. In *Prof. of the IT and KNOWS Conference, XV IFIP World Computer Congress*, August 1998.
- [133] Helena Sofia Pinto and João P. Martins. Ontologies: How can they be built? *Knowl. Inf. Syst.*, 6(4):441–464, 2004.
- [134] Natalya F. Noy and Deborah L. McGuinness. Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, 2001.
- [135] Miran Choi et al. Method and apparatus for social tagging using property field of ontology. United States Patent Application 20100023549, 2010.
- [136] Víctor Rodríguez, Marc Gauvin, and Jaime Delgado. An ontology for the expression of intellectual property entities and relations. In *5th International Workshop on Security in Information Systems (WOSIS)*, June 2007.
- [137] Sergey Krivov, Richard Williams, and Ferdinando Villa. GrOWL: A tool for visualization and editing of OWL ontologies. *Web Semantic*, 5(2):54–57, 2007.
- [138] Marc Gauvin, Jaime Delgado, Víctor Rodríguez, and Eva Rodríguez. OWL Ontology to Represent the DMP Creation Model. Technical report, The Digital Media Project, October 2006.

- [139] Marc Gauvin and Victor Rodriguez. Proposed RRD Requirements. Technical report, The Digital Media Project, January 2007.
- [140] General Assembly 13. Approved Document No. 1 Technical Reference: Value Chain Functions and Requirements, Version 2.1. Technical report, The Digital Media Project, February 2007.
- [141] Marc Gauvin, Jaime Delgado, and Victor Rodriguez. Proposed RRD Text for Approved Document No 3 - Technical Specification: Interoperable DRM Platform, ver. 2.1. Technical report, The Digital Media Project, March 2007.
- [142] General Assembly 15. Interoperable Digital Rights Management Platform. Technical Specification, ver. 3.0. Technical report, The Digital Media Project, July 2007.
- [143] sDae PKU TI BBC, ETRI. AXMEDIS-4HOME Identification of Technological components Required for Apparatus and Ontology Analysis. Technical report, May 2007. [http://www.axmedis.org/documenti/view\\_documenti.php?doc\\_id=2976](http://www.axmedis.org/documenti/view_documenti.php?doc_id=2976).
- [144] sDae PKU TI BBC, ETRI. Specification of AXMEDIS4HOME Architecture. Technical report, July 2007. [http://www.axmedis.org/documenti/view\\_documenti.php?doc\\_id=3251](http://www.axmedis.org/documenti/view_documenti.php?doc_id=3251).
- [145] sDAE. Descripción del Entorno Operativo Común y Casos de Uso. Technical report, Plataforma E2E RM Interoperable, April 2008.
- [146] Marc Gauvin, Jaime Delgado, and Victor Rodriguez. Ontology for IP Media Networks: A response to the MVCO CfP. Input document M15631 for the ISO/IEC JTC1/SC29/WG11, July 2008.
- [147] Marc Gauvin, Jaime Delgado, Víctor Rodríguez Doncel, and Miran Choi (editors). Working Draft of ISO/IEC 21000-19 Media Value Chain Ontology. Technical report, Output document N10070 in the ISO/IEC JTC1/SC29/WG11, Moving Picture Experts Group, July 2008.
- [148] Marc Gauvin, Jaime Delgado, Víctor Rodríguez Doncel, and Miran Choi (editors). Committee Draft of ISO/IEC 21000-19 Media Value Chain Ontology. Technical report, Output document N10264 in the ISO/IEC JTC1/SC29/WG11, Moving Picture Experts Group, October 2008.

- [149] Marc Gauvin, Jaime Delgado, Víctor Rodríguez Doncel, and Miran Choi (editors). Final Committee Draft of ISO/IEC 21000-19 Media Value Chain Ontology. Technical report, Output document N10455 in the ISO/IEC JTC1/SC29/WG11, Moving Picture Experts Group, February 2009.
- [150] Marc Gauvin, Jaime Delgado, Víctor Rodríguez Doncel, and Miran Choi (editors). Final Committee Draft of ISO/IEC 21000-19 Media Value Chain Ontology. Technical report, Output document N10768 in the ISO/IEC JTC1/SC29/WG11, Moving Picture Experts Group, July 2009.
- [151] Marc Gauvin, Jaime Delgado, Víctor Rodríguez Doncel, and Miran Choi (editors). First Draft International Standard of ISO/IEC 21000-19 Media Value Chain Ontology. Technical report, Output document N11146 in the ISO/IEC JTC1/SC29/WG11, Moving Picture Experts Group, January 2010.
- [152] Marc Gauvin, Jaime Delgado, Eva Rodríguez, and Víctor Rodríguez-Doncel. Requirements for a machine readable rights ontology. Technical report, Input document M14707 for the ISO/IEC JTC1/SC29/WG11, Moving Picture Experts Group, July 2007.
- [153] MPEG Requirements Group. Requirements for a Media Value Chain Ontology. Technical report, Output document N9658 of the ISO/IEC JTC1/SC29/WG11, Moving Picture Experts Group, January 2008.
- [154] MPEG Requirements Group. Requirements for a Media Value Chain Ontology. Technical report, Output document N9906 of the ISO/IEC JTC1/SC29/WG11, Moving Picture Experts Group, April 2008.
- [155] Miran Choi, Myung Gil Jang, Hyunki Kim. Proposing an Architecture for MPEG Media Value Chain Ontology. Input document M15698 for the ISO/IEC JTC1/SC29/WG11 , July 2008.
- [156] Jaime Delgado, Eva Rodríguez, and Víctor Rodríguez-Doncel. How to express exclusivity in MPEG-21 REL. Input document M15431 for the ISO/IEC JTC1/SC29/WG11, April 2008.
- [157] Jaime Delgado, Eva Rodríguez, Víctor Rodríguez-Doncel, Marco Balduino, and Franco Bixio. Use of MPEG-21 REL for expression of Audiovisual Contracts. Input document M15430 for the ISO/IEC JTC1/SC29/WG11, April 2008.

- [158] Anna Carreras and Jaime Delgado. A new type of contextual information based on the adaptation authorisation. In *9th Int. Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS08)*, May 2008.
- [159] Vitor Barbosa and M. Teresa Andrade. Multicao: A semantic approach to context-aware adaptation decision taking. In *10th Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS09)*, pages 133–136, May 2009.
- [160] Anna Carreras Eva Rodríguez, Víctor Rodríguez and Jaime Delgado. A digital rights management approach to privacy in online social networks. In *Workshop on Privacy and Protection in Web-based Social Networks (ICAIL09)*, September 2009.