Digital Expression of Agreements on Audiovisual Content Rights

by Víctor Rodríguez Doncel

Submitted in partial fulfilment of the requirements for
the degree of Diploma of Advanced Studies
Doctorate in Computer Science and Digital Communication
Department of Information and Communication Technologies
Universitat Pompeu Fabra

Tutor: Dr. Jaime Delgado Mercé

Barcelona, July 2007

Contents

List of Figures

List of Tables

Chapter 1

Preamble

This chapter presents the whole work, giving a general introduction to the topic and an overview of the document structure.

1.1 Acknowledgements

This document is a milestone in the pursuit of a Philosophy Doctor degree. It tracks the research already done on the field of expression of agreements in the audiovisual market, and it outlines the future work to be done.

The research work has been done under the framework of the Distributed Multimedia Applications Group (DMAG)[1], and also in the context of the Doctorate in Computer Science and Digital Communications held in the Department of Information and Communication Technologies[2] in the Universitat Pompeu Fabra in Barcelona. The work would have not been possible without the cooperation and advice of the DMAG members, specially Jaime Delgado, Silvia Llorente and Eva Rodríguez. I cannot forget mentioning Marc Gauvin, whose points of view have been highly enriching along this time.

1.2 Objectives

This work focuses on the representation of digital agreements in the audiovisual market. Since the Information Technology era started, audiovisual

[1]DMAG, Distributed Multimedia Applications Group, located in the Universitat Pompeu Fabra and the Universitat Politècnica de Catalunya. http://dmag.upf.edu

[2]Department of Information and Communication Technologies in the Universitat Pompeu Fabra, http://www.upf.edu/dtecn

content has been present in a digital form. In the Digital Revolution we live in this era, audiovisual content was very early transposed into the digital world. Internet and the modern telecommunications made possible a fast and efficient exchange of material with Intellectual Property (IP) and an incipient market of digital contents started very soon.

eCommerce had born, and digital audiovisual content was a suitable object of trade. Transactions could be made from computer to computer, with no physical presence. Catalog browsing was possible from the computer, safe payment systems were available from the beginning and content delivery of the contents could be done entirely through the network.

Major efforts have been focused since then on the distribution of final products to the consumer, and the digital B2C market has acquired a mature development. However, Intellectual Property Objects are object of trade not only for the final consumer. From the very original idea in an author´s mind, until the final product, there have been some other intermediate IP objects along this process (that we call *value chain*), and they are subject of a possible trade too. In this B2B sector the regulations and agreements have remained up to date in the analogue world.

The major objectives of this work are to analyze the Intellectual Property Objects along its value chain, and to express digitally their governing agreements.

A good expression of the agreements in a digital format would permit better Digital Rights Management (DRM) systems. And while the idea of *digital license*, expressed in a Rights Expression Languages (REL) to govern a DRM system is not new, it is the aim of this work extending this digital license so that all the processes in the value chain are covered, and it is the aim expressing this license at a semantic level, rather than the syntactic level of the RELs.

1.3 Methodology and plan

In this document we will view how the agreements in the audiovisual market are along its complete value chain, and how these agreements could be digitally represented. This digital representation can be done at a syntactic level, like the existing REL languages, or at semantic level based on a computer ontology. We will review these REL formats as well as the enabling technologies for implementing the semantic approach. With all these ingredients, we will be on the way of realizing the innovative idea of representing semantically contracts and taking advantage of its potential.

Chapter 2

Introduction

It is a claimed objective of this document offering a representation of the business model of the audiovisual content market. But before discussing how this representation should be, the model itself must be clearly described.

Modelling a problem is by itself a key task, and requires making a generalization of the particular elements that appear recurrently. This abstraction process is of critical importance, and although a starting model was considered (that proposed by the DMP group[1]), a thorough review has been done.

2.1 Intellectual Property

We call *Intellectual Property Objects* to those Objects over which *Intellectual Property Rights* apply. Intellectual Property Rights are the extension of the concept of property right from material objects to abstract ideas. In the same way a man is owner of his material properties and can exercise rights over them, a man is also owner of its intellectual creations and has equivalent rights.

It is not the purpose of this document to adhere to any particular legal system. Each jurisdiction has its own corpus of laws and in many aspects, some are contradictory between themselves. But there is a general acknowledgment of the Intellectual Property concept and the general protection that covers the authors of original ideas. The foundations of the ideas developed in this document are based on this common substrate that is universally agreed by the different legal bodies.

Nowadays some groups are raising their voices for the abolition of the concept of Intellectual Property at all. One of the most prominent repre-

[1] The Digital Media Project, http://www.dmpf.org

sentants of this movement is Richard Stallman, who claims that Intellectual Property should be abolished based on the idea that making a copy of an intellectual creation does not reduce the value of the original manifestation. While the free discussion of different ideas generates by itself an enriching debate, so far no legal system has backed their theses and for the rest of the document they will be ignored. Naturally, the controversial on the protection technics of Intellectual Property Objects implemented in current DRM systems will be avoided as it is of no direct interest for the purposes of this work.

2.2 The Intellectual Property Value Chain Model

Having defined what IP Objects are, we shall give an overview of its value chain. Because being far from being static, ideas evolve, they mix naturally with other ideas and refine themselves until they reach a final form. Here the agreed value chain model will be shown as described by DMP in [1].

We will call *work* to the abstract raw idea that will be attributed to a single creator. Creators can be in fact groups of persons, but we can treat them as a single entity and then the concept remains unchanged. The creator of the idea is sometimes referred as *author*, but here he will be called simply *creator*.

Ideas are base for other ideas, and when this dependency is manifest, then we say that a work is an *adaptation* of another identified work. *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. Note that while *works* or *adaptations* are mere conceptual entities, manifestations are physical objects.

Music compositions, theater pieces and other IP Objects are susceptible of being played, interpreted or performed. This play, interpretation or execution will be known in this document as *instance*, and the person who carries it out will be called *instantiator*. An *instance* is therefore an event, that can only happen once. If this session is somehow recorded (with a video cam, an audio recorder or any other means), it is said that the instance has been *fixed*. It is also referred as the *first fixation*.

This first fixation will be the material that a producer will take for making *copies* arranging them properly. Finally, collection of copies making *products* will be distributed for an *end user* to enjoy it.

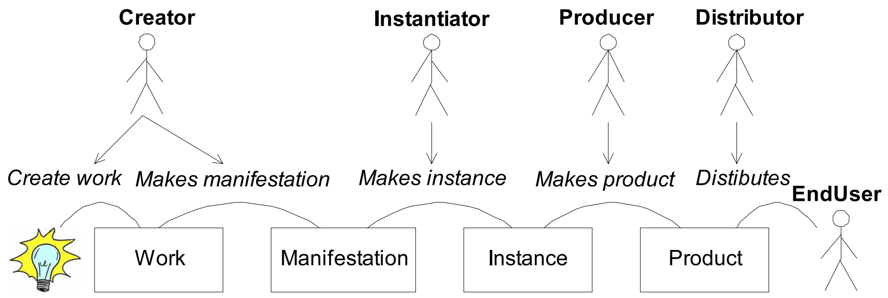


Figure 2.1: The IP Value Chain.

Figure 2.1 shows three categories of elements. Some persons (*roles*) 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 pass 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 can be of 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.

2.2.1 IP Entities

IP Entities are one of the three basic categories of entities the model deals with. The relationship between the upper IP Entities can be described with the Figure 2.2.

The enumeration of different IP Objects we can recognize in this process is 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.

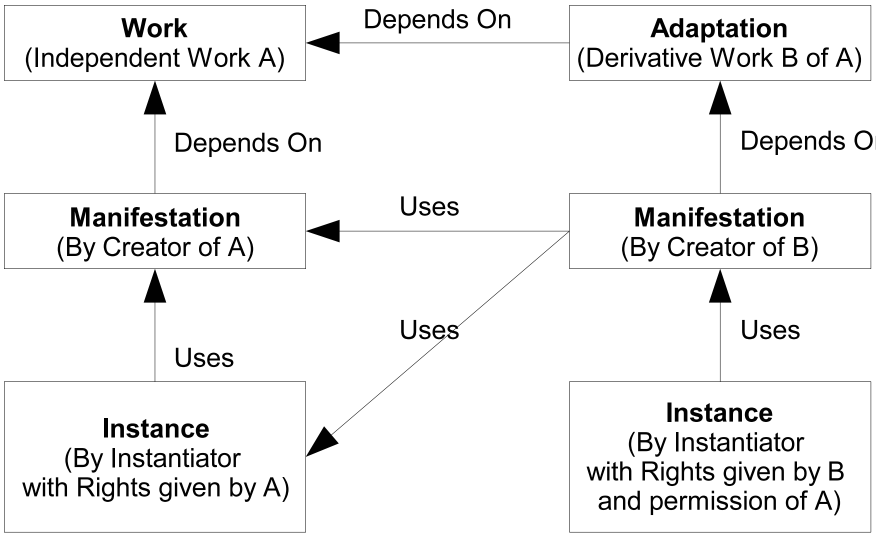


Figure 2.2: Basic IP Model

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.

Each of the previous 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 *Work-Manifestation* and *AdaptationManifestation* respectively etc. Each of these concepts will be further detailed in the next sections.

Figure 2.3 extends Figure 2.2, including different the refinement of these objects. Additionally, it shows the *Copy* IP Entities, whose origin can be a manifestation or an instance.

Some basic relations can be defined between these IP Entities: a *dependsOn* relationship links natural dependency as between a manifestation and a work, and the relationship *uses* links a copy and an instantiation. *dependsOn* would pose a logical relationship, while *uses* involves more a physical business.

2.2.2 Roles

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

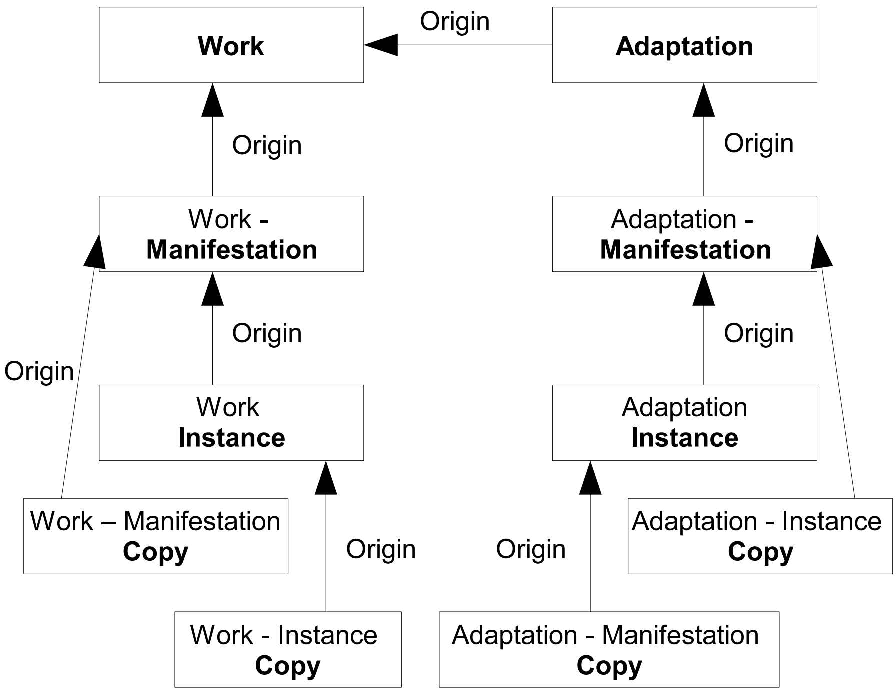


Figure 2.3: Refined IP Model

group thereof who incarnates one or more roles. Roles are the second big category of entities this model deals with. The list of roles is the following:

Creator. [2] The author of the work, who translates his idea into a material realization.

Adaptor. [3] The creator of an adaptation from a work.

Instantiator. [4] An agent who executes a performance or rendition of the work.

Producer. An agent who compiles commercial distributable products.

Distributor. [5] An agent who distributes the product.

End User. The last agent to use the content.

The basic *leitmotiv* of this document 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.

2.2.3 Actions

The creator has full rights over the work, and he can trade with these rights. The execution of the rights is called *actions*. The actions that can be performed on or with the IP Entities can be grouped into:

- Actions that generate new IP Entities: create, adapt, make manifestation, make instance, make product..

[2] 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.*

[3] 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.*

[4] 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.*

[5] 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.*

- Actions required to use IP Entities , like the act of "playing" a song etc.

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. This is an advance from what will be extended in forthcoming sections.

The conceptual model has also a clear idea about how rights are traded. It says that 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 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.

In the next sections, this model will be precised and developed into its details.

Part I

State of the Art

This part reviews the state of the art needed to carry out the work. The knowledge areas this document touches are broad, including both technological and legal fields.

This part will describe first what contracts are, and how they are understood in this work. The classification of contracts in *passive* (chapter 3) and *active* (chapter 4) that has been done here, is rather unconventional. However, for the sake of our purposes it is more convenient to separe agreements depending on if they are mere reference texts (passive agreements) or a driving part in the commerce process (active agreements).

As the knowledge upon which this work is based has also a strong technological foundation, Chapter 5 will summarize what is needed to represent semantically the contracts along the value chain.

Chapter 3

Passive representation of agreements

An *agreement* is a mutual promise between two or more parties. A *contract* is the representation of the agreement. In this chapter the evolution of the contracts through the ages will be reviewed, giving roots and sense to the aims of this work.

3.1 Classical Contracts

Agreements have been written since several millennia ago. Documented agreements appear, in fact, in the earliest stages of the writing development, showing how important has always been leaving permanent testament of the agreed terms. They had quite a simple task: having an undisputed authority reminding the contract terms in case of dispute.

The Hammurabi code, carved in diorite stone at some time thirty centuries ago, is the first remarkable public law document we keep. Personal agreements, belonging to the private law documents sphere, appeared roughly at the same time. Ordered trade fostered economical growth and civilization that adopted written documents developed much faster. Indeed, this work does not focus on juridical sciences or stone documents, but the author would like to stress that the same principle still rules, 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.

Contracts from the very early times to the threshold of the XXth century have been classified as *classical contracts*. Contracts gained in complexity

Figure 3.1: One of the very first contracts we keep.

through the centuries, but essentially remained stable in objectives and style until the XXth century. The major disruption in the contract tradition that happened in the last century has to do with the scope of the contract. Classical contracts focused only on a single event. That event could be a marriage (subject to a contract), an inheritance or the purchase of a house or a land, and in such a sense we can qualify them as *static contracts*. The paradigm at the center of classical contract law was a snapshot taken at the moment a bargain was made. As will be seen, this does not correspond with the current scope of the contracts we are interested in, where rather than an event, a process is tracked, and royalties amount is calculated dynamically and based on a complex number of dynamic elements etc. It is rather instructive following the evolution of contracts complexity pointed out by Eisenberg in [2]. But it is also instructive finding out the common features that link these primitive contracts to those we deal with here.

3.2 Modern Narrative Contracts

Paper narrative contracts

The twentieth century witnessed the development of a modern contract law that largely overthrew classical contract law. Contracts became *dynamic*, and 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; the static legal-duty rule had withered almost completely away, to be largely replaced by a dynamic modification regime that took into account the value of ongoing reciprocity. A static review of liquidated damages provisions was giving way to a dynamic review that took account of the actual loss; and static offer-and-acceptance rules were replaced by dynamic rules, such as the duty to negotiate in good faith. Nevertheless, a human judge was at last the ultimate responsible of interpreting the terms of the contract, what will contrast with what is presented in the forthcoming sections.

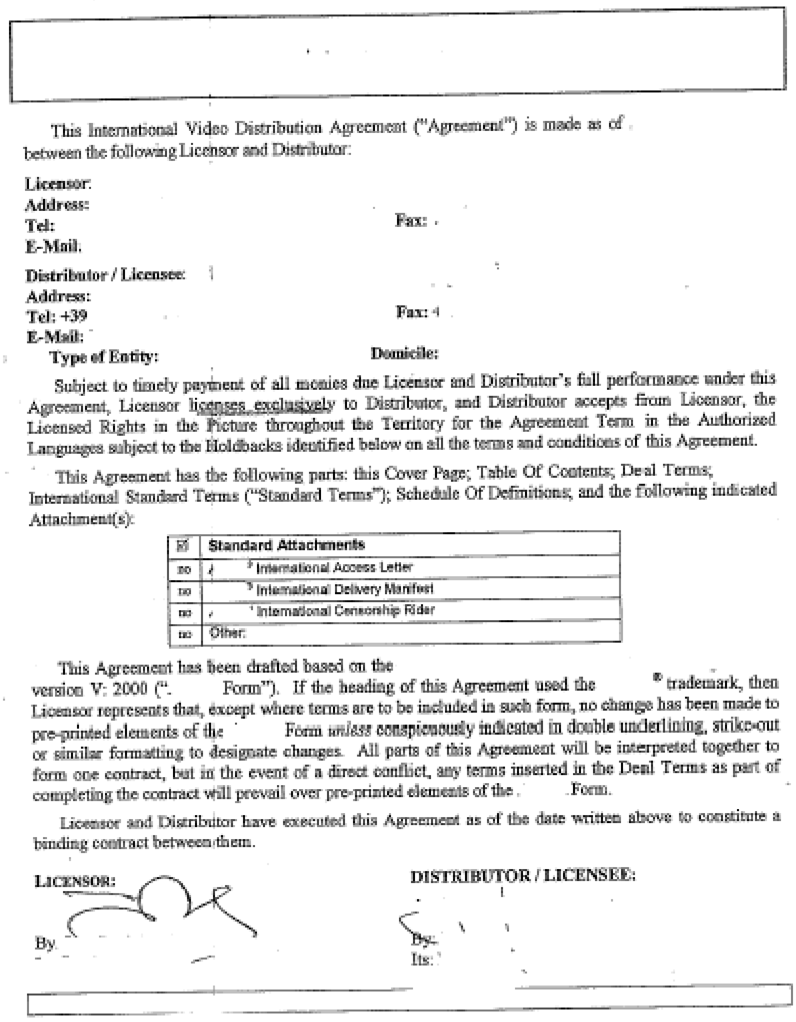
Electronic narrative contracts

The last quarter of the XXth century and its Digital Revolution did not bring any essentially new change. By *narrative contracts* we refer to those documents expressed for the human intellect, no matter if they are written with cuneiform symbols or ASCII characters. They are simply a representation of the human language and so the first narrative contracts in digital formats were still narrative contracts.

Electronic representation of contracts brought undoubtedly many benefits, as it was much easier and better storing a WordPerfect[1] file in a diskette than keeping it in a paper archive. This change still fulfilled all the objectives of narrative contracts, well defined in [3]:

- *To act as a guarantee*, signature being a key element to avoid repudiation. Note that electronic representation of narrative contracts replace the signature with the digital signature, of equivalent functionality with recognized juridical value.
- *To stay as an enduring material base for the will declaration.* (spoken word is easily forgotten). Again, electronic representation of narrative contracts is also stored in a physical medium, able to be kept as any other object.
- *To act as a probative element in court trials.* It has been formally recognized as such in times as remote as the ancient Greeks age. And digital contracts have acquired also the same juridical acknowledgment. Note that without this acknowledgement, Digital Right Management (DRM) would have been rather less useful.

[1]Wordperfect is a trademark of Novell Inc.



This International Video Distribution Agreement ("Agreement") is made as of between the following Licensor and Distributor:

Licensor:
Address:
Tel: **Fax:**
E-Mail:

Distributor / Licensee:
Address:
Tel: +39 **Fax:**
E-Mail:
Type of Entity: **Domicile:**

Subject to timely payment of all monies due Licensor and Distributor's full performance under this Agreement, Licensor licenses exclusively to Distributor, and Distributor accepts from Licensor, the Licensed Rights in the Picture throughout the Territory for the Agreement Term in the Authorized Languages subject to the Holdbacks identified below on all the terms and conditions of this Agreement.

This Agreement has the following parts: this Cover Page; Table Of Contents; Deal Terms; International Standard Terms ("Standard Terms"); Schedule Of Definitions; and the following indicated Attachment(s):

☑	Standard Attachments
no	International Access Letter
no	International Delivery Manifest
no	International Censorship Rider
no	Other:

This Agreement has been drafted based on the version V: 2000 (". Form"). If the heading of this Agreement used the ® trademark, then Licensor represents that, except where terms are to be included in such form, no change has been made to pre-printed elements of the Form *unless* conspicuously indicated in double underlining, strike-out or similar formatting to designate changes. All parts of this Agreement will be interpreted together to form one contract, but in the event of a direct conflict, any terms inserted in the Deal Terms as part of completing the contract will prevail over pre-printed elements of the Form.

Licensor and Distributor have executed this Agreement as of the date written above to constitute a binding contract between them.

LICENSOR: **DISTRIBUTOR / LICENSEE:**

By By:
Its:

Figure 3.2: Narrative contract in the audiovisual market.

3.3 Contracts in the Audiovisual Market

Contracts in the audiovisual market are those that deal with the transfer of certain rights over resources which are Intellectual Property Objects.

They usually exhibit the same pattern of characteristics. Most of the times, they are agreements between only two parties, and the parties belong to a reduced set of characters: those defined in section 2.2. More specifically, contracts are usually either addressed to an intermediate artist or to a distributor. Contracts addressed to other content creators are for allowing them to include or adapt already protected material, contracts addressed to distributors include a wide range of channels of distribution for the content to be played.

3.3.1 Legal Framework of the Public Law

But although by definition contracts belong to the sphere of private law, they cannot infringe what the public law says about the matter. For example, a contract may say an author will refuse creating any artistic work for a certain period, but this contract would be not valid as this clause is declared illegal according to the law[2].

Intellectual Property Rights have been often classified as Industrial Property Rights and Copyright, being the later the one which concerns us. Public law is quite homogeneous across the different national regulations, as most of the countries, or at least the vast majority of developed countries, adhere to the same set of agreements on Industrial Property Rights and Copyrights. A brief review can be found on [4].

These agreements have had a long history, with the concept of copyright in the Anglo-Saxon legal systems and the slightly different French concept of *droit d'auteur*. They were for the first time universally agreed in the *Berne Convention for the Protection of Literary and Artistic Works*, usually known as the Berne Convention, held in Berne, Switzerland in 1886.

In the seventies of the XXth century, interest on these topics grew and a new series of international treaties were signed. This is not a surprise at all. If we are truly living a Digital Revolution, the center of interest moves from manufactured products to information and knowledge, and what these agreements try to regulate is information commerce (artistic, industrial or any other). In fact, they were brought into the center of the scene in the 1994 meeting of the General Agreement on Tariffs and Trade (GATT) (later replace by the World Trade Organization, WTO) in Uruguay, when

[2]For example, according to the Spanish Ley de Propiedad Intelectual, art. 43.3

the TRIPS (Agreement on Trade Related Aspects of Intellectual Property Rights) agreements were signed.

WIPO Agreements

These and other agreements were updated and summarized in the pivotal WIPO[3] agreements of 1996 [5]. They deal with the rights of authors, performers (*instantiators* in the vocabulary of this document), producers and distributors, that is to say, they cover the whole value chain we have described. WIPO agreements were two:

WIPO Copyright Treaty (WCT) [4] , concerning author rights, and updates the previous *Berne Convention for the Protection of Literary and Artistic Works* (1971).

WIPO Performance and Phonogram Treaty (WPPT) [5], mainly concerning performers, producers and distributors rights. It supersedes the *Rome Convention for the Protection of Performers, Producers of Phonogams and Broadcasting Organisations* (1961) and the *Geneva Convention, for the Protection of Producers of Phonograms against unauthorized Duplication of their Phonograms* (1971).

It took six additional years until it was signed by 30 member countries, the minimum number that the UN considers in order to make its application effective. Nowadays it is universally in force.

WCT was implemented in the US in the 1998 through the important law known as Digital Millennium Copyright Act (DMCA) [6]. DMCA criminalizes 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.

Three years after WIPO treaties were also approved in Europe by the European Council as the *European Union Copyright Directive* (2001/29/CE) [7], developed in two additional directives. According to these new directions, the EU member states adapted their national laws.

Spanish legal framework

For the particular case of Spain, which will be treated with special attention later on this document, the law regulating Intellectual Property is the *Ley de*

[3]World Intellectual International Property Organisation (WIPO), http://www.wipo.int
[4]WCT Treaty: http://www.wipo.int/treaties/en/ip/wct/trtdocs_wo033.html
[5]WPPT Treaty: http://www.wipo.int/treaties/en/ip/wppt/trtdocs_wo034.html

Propiedad Intelectual (LPI) dated back originally on 11th November 1987. It 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, 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).

It should also be mentioned the known as *Ley de Internet*, 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 our interest, it is enough to know that it enables the existence of electronic commerce by acknowledging electronic contracts as full valid contracts (Art. 23).

Chapter 4

Active representation of agreements

Encyclopedia Britannica defines contract as "a promise enforceable by law", Nowadays, digital contracts can give a step further and make the enforcement *themselves*.

As it has been said in the previous section, electronic contracts are written in a language understood by computers, and thus computers can interpret it and play an active role. It will be in the context of Information Technology systems where digital contracts or licenses will authorize or deny the access to resources. This is in fact a central idea of the DRM (Digital Right Management). DRM does not substitute the justice and the courts, but it is a complement.

DRM refers to the technologies used by publishers or copyright holders to control access to or usage of digital data or hardware, as well as to restrictions associated with devices or the protected resources. The information of which users can exercise which rights over which resources is what constitutes the *licenses*.

The use in this document of the words *contract* and *license* is precise and has to be well defined. From a juridical point of view, *contract* revolves around the notion that two or more parties have bargained or negotiated an exchange of promises, while *license* is simply the permission to do an act that, without the permission, would be unlawful. Licenses are not subject to counter proposals.

However, licenses in the REL jargon, are containers of rights regardless their negotiation. In this document, *license* will be the document written in a Rights Expression Language to manage a DRM. Licenses were designed with this purpose in mind. And *digital contract* will be the representation of a narrative contract, regardless of its use for managing a DRM or not. Digital contracts are not expressly designed to govern a DRM (although in occasions they can).

In this section state of the art in expressing electronic licenses and electronic contracts will be detailed.

4.1 RELs and Electronic Licenses

License is then the document written in a Rights Expression Language as a part of a Digital Rights Management system. This section will review their brief and unfinished history. It is unfinished because first RELs were developed in the late 1990's, but none can be considered to be fully deployed up to this date (as of 2007).

4.1.1 MPEG 21 REL

Origin

The development of today's rights expression languages started with the work of a Xerox PARC's scientist 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 the trusted system software, and so he started working on the Digital Property Rights Language (DPRL).

DPRL appeared in a patent filed by Xerox in November of 1994 (and granted in February of 1998) [8]. In November of 1998, Xerox issued the first XML version of the Digital Property Rights Language (DPRL), labeled Version 2.0. Prior to that time, DPRL had been written in the LISP programming language.

XrML

In 1998, version 2 of DPRL was licensed to a new company founded by Microsoft and Xerox called ContentGuard[1], which developed DPRL into the eXtensible Rights Markup Language (XrML) [9].

XrML structure is composed of an *issuer*, and his issued *grants*. His issued grants hold elements to describe the *principal*, the *resource*, the *rights*

[1] ContentGuard, http://www.contentguard.com

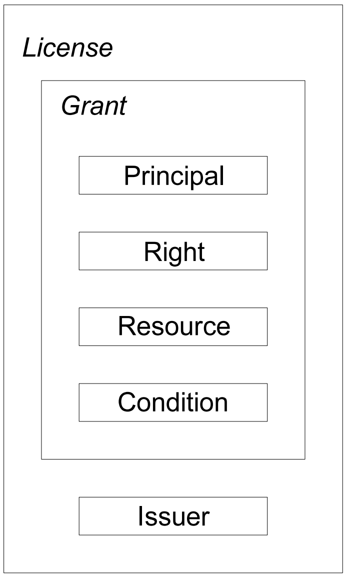


Figure 4.1: The XrML license.

and the *conditions*. A license is conceptually a container of *grants*, and grants allow a *principal* to exercise *rights* against a *resource*. An schema of the license can be seen in Figure 4.1.

Version 1 of XrML was published in 2001. This time included new features describing some methods to make the REL robust. It added 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 the 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 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.

In 2003, XrML was used as the basis for the rights expression language for the MPEG-21 standard, and its basic structure lasted.

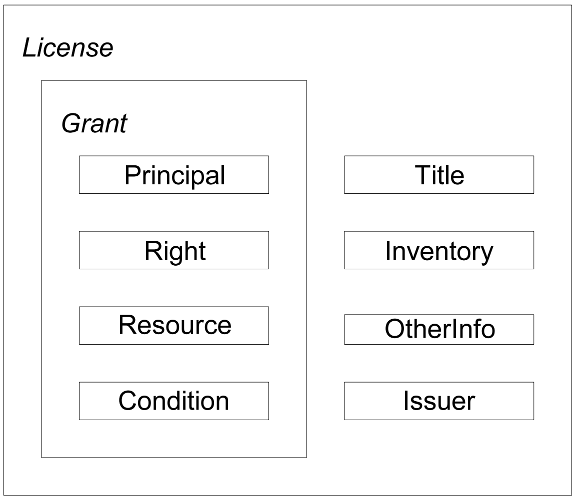


Figure 4.2: The MPEG-21 REL license.

MPEG-21 REL

On April 2004, MPEG-21 Part 5 was published as an ISO Standard [10]. ISO 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.'*

Hardly one month later, a complementary standard was approved as ISO Standard [11], 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 is largely based on the XrML 2.0, and the structure of the license is similar to the one shown in Figure 4.1. The elements of the license not described before are:

Title: A descriptive phrase about the License that is intended for human consumption in user interfaces.

Inventory: Defining variables is possible within a license with this element. This is a syntactic mechanism for reducing redundancy and verbosity in Licenses that can be used throughout a License.

GrantGroup: It is a container of Grants.

Part	*Namespace prefix*	*Namespace*
Core	r	urn:mpeg:mpeg21:2003:01-REL-R-NS
Standard	sx	urn:mpeg:mpeg21:2003:01-REL-SX-NS
Multimedia	mx	urn:mpeg:mpeg21:2003:01-REL-MX-NS

Table 4.1: Namespaces prefixes

Grant: The authorisations, holding resources, principals, rights and conditions.

Other information: An element to be filled in freely.

As it has been said, a grant is formed by four elements. The *principal* represents the unique identification of an entity involved in the granting or exercising of Rights. The *right* specifies an action or activity that a *principal* may perform on, or using, some associated target *resource*. The *resource* represents the object against which the *principal* of a *grant* has the *right* to perform. The use of a digital resource in a *grant* provides a means by which a sequence of digital bits can be identified within the *grant*. The *condition* element represents grammatical terms, conditions and obligations that a *principal* must satisfy before it may take advantage of an authorization conveyed to it in a *grant*.

For implementing this, three schemes are given. The REL Core Schema is equivalent of the XrML version 2 and defines the structure of the license, while the REL Standard Extension and the REL Content Extension provide the useful elements such as practical rights and conditions. Since the standard was published, some other extensions have been added. The three initial namespaces with its usual prefix are shown in Table 6.3.2:

4.1.2 ODRL

ODRL (Open Digital Rights Language [12]) is also an XML-based standard Rights Expression Language (REL), and it was conceived in 1997 by John S. Erickson and Renato Ianella. 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[2] (OMA).

[2]OMA, Open Mobile Alliance, http://www.openmobilealliance.org/

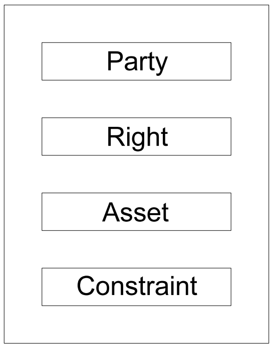


Figure 4.3: The ODRL license.

Actually ODRL is managed by an open organization that is 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[3]. There is at least one open source implementation of ODRL available.

ODRL defines *parties* (users and Rights Holders), *assets* (any physical or digital content), and *rights* that include *permissions* (the usages or activities allowed over the users). The *permissions* can contain *constraints*, *requirements* and *conditions*. The difference between constraints and requirements is that constraints are limits to the permissions, and requirements are obligations that have to be fulfilled, while conditions are used to specify exceptions. Figure 4.3 shows the ODRL license structure.

4.1.3 Creative Commons

First developed in 2002, Creative Commons[4] (CC) provides an expression of rights for open access web resources, including HTML documents and digital audio files. The CC license is machine-readable in the sense that it is in the form of a digital document, but there is no machine-actionable control over use of the content that carries such a license. In fact, the CC 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. Figure 4.4 shows the symbols for "attribution", "noncommercial", "no derivative works" and "share alike" conditions. The CC metadata record includes a

[3]The Dublin Core Metadata Element Set is a vocabulary of fifteen properties for use in resource description. http://dublincore.org/

[4]Creative Commons, http://creativecommons.org/

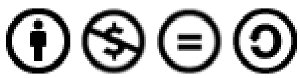


Figure 4.4: Creative Commons symbols that determine the license
.

section with Dublin Core metadata elements to describe the resource.

It is very easy to be created, and perhaps for this has become very popular for those who want to grant their own work for free at least only until some extent. But it is of few interest for the purposes of this work, as their content is not intended to be read by machines, and it relies and trusts exclusively in the prevailing copyright law to protect the digital content.

4.2 CELs and Electronic Contracts

Electronic contracts are not intended merely to make a DRM system work, but they have played an active role in electronic commerce far earlier than the licenses we have seen in Section 4.1. In this section we will review briefly the electronic contracts.

4.2.1 Early electronic contracts representation

The earliest electronic contract representation were born together with the electronic commerce and the first Electronic Data Interchange (EDI) standards. EDI has been of huge importance in the industry, and it comprises a set of standards for structuring information to be electronically exchanged between and within businesses, organizations, government entities and other groups.

COSMOS [13] was an e-commerce architecture developed in the 1996 supporting catalog browsing, contract negotiation and contract execution. It defined a contract model in UML and proposed a software architecture CORBA based in a quite coherent manner. For our interest, it has to be remarked their use of UML and its high expressivity. Actually, an UML specification could somehow be seen as a Computer Ontology [14], if the models had a standard representation.

DocLog [15] was an electronic contract representation language introduced in the 2000. The contract was an 'XML like' document with information structured in tags but not compliant with XML. After its presentation it has not been an active project.

Business Contract Language [16] (BCL) was a contract language specifically developed to express contract conditions for run time contract monitoring. BCL is expressed in two syntactic forms. In the running system, a portable XML-based notation is used, but there is also a human-readable language with more conventional syntax. The human-readable version is generated automatically from the XML system. Although BCL focuses on monitoring rather than enforcing, we can consider it a meaningful precedent. In the same line but from 2005, [17] also studies electronic contracts and its negotiation and enactment through their workflow.

Further XML based contracts representation appeared in the coming years, like the Secure Contracts system [18]. This system also tracked the whole process of products offering, contract negotiation and contract execution, reflecting this in the contract structure. This contract structure held the contract itself (parties, resource, conditions, legal terms), and also status information and log information to track its execution. As interesting feature, it was made compliant with the IOTP (Internet Open Trade Protocol). The IOTP is an Internet protocol (RFC 2801), which describes *The negotiation of who will be the parties to the trade, how it will be conducted, the presentment of an offer, the method of payment, the provision of a payment receipt, the delivery of goods and the receipt of goods.*

4.2.2 CEL: Contract Expression Language

CEL (Contract Expression Language) [19] is an initiative to represent contracts also in XML, dating from January 2004 and proposed by the Content Reference Forum[5]. It is quite logical and does not depart far from MPEG-21 REL, with whom shares many features.

It is defined as follows: *'an XML-based language to express contractual agreements between business entities.'* They claim as objective to govern the distribution and use of content, while keeping human readability: *'the syntax and semantics of its expressions are suited for both human and machine interpretations'*. The CEL is based on the same architecture model and it shares many basic elements with the MPEG-21 REL.

For the purposes of this work, it should be highlighted its bright approach 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. Taking

[5]Content Reference Forum, http://www.crforum.org

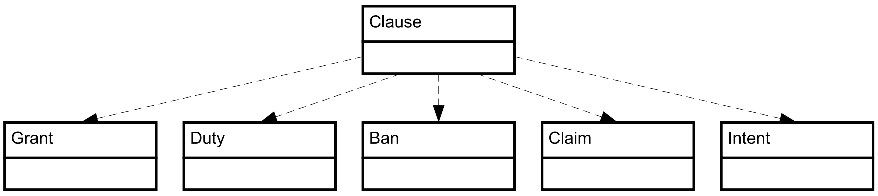


Figure 4.5: Classification of clauses in the CEL model.

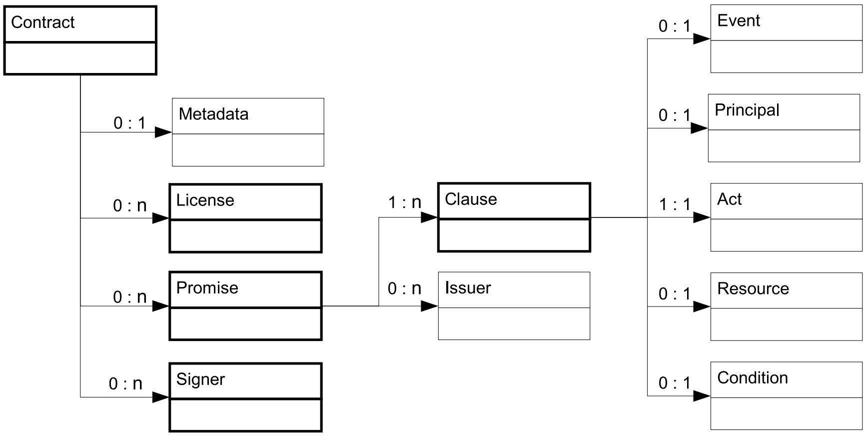


Figure 4.6: The CEL contract model.

this into account, CEL clauses are classified as follows:

A more detailed analysis in deontic logic for contracts can be found in [20].

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*. The following figure shows the structure of a contract expressed in the CEL language. It is very interesting the idea of modeling *events* in the contract, and the work here presented will take this idea from here.

A sample contract is shown with the 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="dea" </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 much projects are considering it. The next paragraphs describe a new and more promising alternative.

4.2.3 eContracts

The Business Narrative Markup Language Schema was an electronic business documents representation developed by a private company. It was an XML based approach, well structured and with very few defined elements. It was the base upon which eContracts was designed.

eContracts [21] is the name of the OASIS[6] LegalXML eContracts Technical Committee (TC) that closed on May 2007 the electronic contracts standards of the same name. The eContracts Schema is intended to describe the generic hierarchical structure of a wide range of contract documents (including audiovisual contracts). The TC envisaged that the primary use of the eContracts Schema was to facilitate the maintenance of precedent or template contract documents and contract terms by persons who wished to use them to create new contract documents with automated tools. Use cases covered include negotiated business contracts, ticket contracts, standard form business and consumer contracts and click-through agreements.

The eContracts Schema is provided in Relax NG[7] compact syntax (being normative), XML Schema[8] (XSD) and as a DTD [9]. The eContracts Schema

[6]OASIS, Organization for the Advancement of Structured Information Standards, http://www.oasis-open.org/

[7]Relax NG is a schema language for XML. http://relaxng.org/

[8]XML Schema http://www.w3.org/XML/Schema is a schema language

[9]Document Type Definition (DTD) describes the structure and syntax of an SGML or XML document, http://www.w3.org/XML/1998/06/xmlspec-report.htm

uses features that cannot be represented in DTD syntax, as described in the eContracts specification, so DTD does not provide a complete specification.

Having appeared so recently, there is currently no open source or commercial software that provides ready to use transformation or publishing applications for the eContracts Schema, but they are likely to appear.

The eContracts Core Schema is simple and defines only 53 elements. As the reader may guess, *contract* is the root tag for contracts (and like in CEL). A contract must have a *title* and a *body*, and may contain a *contract front*, *metadata*,*back* and *attachments*.

The *body* part of a contract will consist of numbered *items*. In some cases, these items may be preceded by one or more *blocks* representing introductory paragraphs, *blocks* represent grammatical or structural paragraphs. The *item* is the basic building block of the document hierarchy. It is a recursive element and represents structures that is found in contracts as *chapters*, *parts*, *sections*, *clauses* and *subclauses*.

A sample contract follows:

```
<?xml version="1.0" encoding="utf-8"?>
<contract xmlns="urn:oasis:names:tc:eContracts:1:0"
xmlns:dc="http://purl.org/dc/elements/1.1/">
<metadata>
<dc:title>Sample contract</dc:title>
<dc:creator>Victor Rodriguez</dc:creator>
</metadata>
<title><text>Example of contract</text></title>
<subtitle>On the transfer or rights of this DEA</subtitle>
<contract-front>
<date-block>
<date><em>2007</em></date>
</date-block>
<parties>
<party><person-record><name>Bob</name></person-record></party>
<party><person-record><name>Alice</name></person-record></party>
</parties>
</contract-front>
<body>
<block>
<text>Bob will allow Alice adapt this DEA.</text>
</block>
</body>
<back></back>
<attachments>
<attachment> </attachment>
```

```
</attachments>
</contract>
```

Chapter 5

Enabling technologies

In our context we understand by *enabling technologies* those technics that make possible a semantic representation of agreements to be developed and work within the framework of a DRM system.

There is a number of enabling technologies that are almost invisibly assumed to be at hand, and yet they are crucial. This is the case of digital signature and fingerprinting of the contracts, which provide them some properties like authentication, integrity and non repudiation, or URIs[1], which identify resources without conflicts. These and other technologies shall not be commented in this work, instead, only those directly related to knowledge representation will be handled.

5.1 Knowledge Representation as Computer Ontologies

Knowledge Representation aims at making knowledge as explicit as possible. It is a multidisciplinary field that applies theories and techniques from logic (providing the formal structure and rules of inference), traditional Ontology theory (defining the kinds of things that exist in the application domain) and computation (supporting the applications that distinguish knowledge representation from pure philosophy).

There has been several representation languages considered as computer ontologies, and among them this work focuses on OWL (Ontology Web Language), which is traced here from its family. For a review of other ontology formats, [22] compares different alternatives.

[1]Uniform Resource Identifier (URI), is a string of characters used to identify a resource. (RFC 2396)

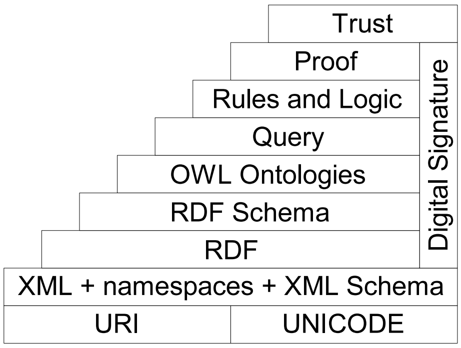


Figure 5.1: The Semantic Web Stack of Technologies

5.1.1 The Semantic Web Stack of Technologies

The OWL is only one step in the Semantic Web stack of technologies. Figure 5.1 depicts the schema proposed by Tim Berners Lee (the creator of the WWW and the Semantic Web).

URI is the well known universal, web based, naming scheme, and Unicode the standard character encoding, upon which XML together with namespaces and XML Schema can be built. They provide a syntactic schema, basis for the RDF which describes well resources and metadata. RDF Schema and OWL are able to represent the semantics of the data, over which queries can be launched. The results of these queries with the proper rules of the logic can proof the results; if the data is digitally signed then *trust* will be achieved.

Until the XML layer the technologies are of general interest and will not be discussed, the upper layers will be here commented.

5.1.2 RDF

The Resource Description Framework (RDF) [23] is a standard (technically a W3C Recommendation) for describing resources (a resource being anything we can identify). It can be seen as an standard for declaring sentences, sentences always expressed in its most simple form from a linguistic point of view: subject, verb, object. In RDF language, though, they take a different name.

Each sentence in an RDF Model is called a *statement* or a *triple*. Each statement asserts a fact about a resource, and can be represented with an arc as in Figure 5.2 (in italics, the RDF terminology).

The three parts of the statement are:

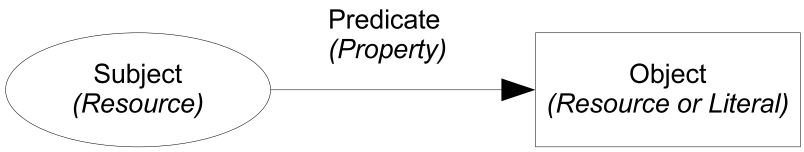


Figure 5.2: RDF triple.

- The subject is the *resource* from which the arc leaves
- The predicate is the *property* that labels the arc
- The object is the *resource* or *literal* pointed to by the arc

5.1.3 RDF Schema

RDF Schema (RDFS) [24] is the semantic extension of RDF. It is a primitive ontology language (its specification started in 1998), with the basic elements for specifying a complete semantics by extending RDF. It defines relations like *subClassOf*, or a domain and range for the properties etc., and defines a class for all the *Properties*, *Resources* etc.

5.1.4 OWL

Web Ontology Language, or OWL, is a W3C Recommendation [25] intended to "explicitly represent the meaning of terms in vocabularies and the relationships between those terms". In addition to defining hierarchical classes that resources can belong to, OWL allows the characteristics of resource properties to be expressed.

OWL could be used to state that the *childOf* property is the inverse of the *parentOf* property, or that *brotherOf* is a transitive property, while this is not possible with just plain RDF or RDFs.

OWL components

The key concepts that lie under OWL and other Ontologies are *classes* and their relations expressed as *properties*.

- Classes: These are hierarchically organized, so that concepts can be further specialized from more general ones in the form of subclasses and individuals. For example, the class *Broadcast* may be seen as a specialization of the more general class *PublicCommunication*. This would be done using the inherent RDF relator "subclassOf". The Ontology defines, when pertinent, which classes are disjoint i.e. individuals belonging to a class are prevented from simultaneously belonging to another specified class. For each class, the necessary and/or sufficient conditions for the specialization to be valid are given.

- Properties: OWL defines three kinds of Properties that any class may possess, the so called *object*, *datatype* and *annotation* properties. At this stage we are mostly concerned with object properties. Object properties are the most interesting since they allow us to define the relationship between individuals. Datatype properties relate individuals to specific datatypes e.g. birthday, has the data type property "date".

- Individuals: The ontology can also include class individuals as a part of its definition, but in this ontology there has been no individuals defined.

OWL complexity levels

OWL is comprised of three sub-languages, OWL Lite, OWL DL and OWL Full, of increasing complexity. The ontology described in this document adheres to OWL DL (Domain Logic), which is essentially a First Order Logic. The Logic would be of a higher level, if assertions where the resources are other statements were raised. Upper Order ontologies make harder the reasoning tasks and pose a number of theoretical problems.

Apart from this complexity from a formal point of view, there is an inherent complexity in RDF documents that prevents it from being so *human readable*. It is because statements (or triples) are the only ingredient of OWL, and what is one hand good (it is a simple format) it is bad in the other(complex ideas have to be decomposed in simple statements). For example, to express the logical sentence *Johnny´s parents are Alice and Bob*, something similar to the next sentences have to be internally created:

```
AnonymousClass0 isA Set
AnonymousClass0 startsWith Alice
AnonymousClass0 continuesWith AnonymousClass1
AnonymousClass1 startsWith Bob
AnonymousClass1 continuesWith NULL
Johnny hasAsParents AnonymousClass0
```

All the sentences are simple triples, but there the format looses in clarity, at least from the human point of view. This artificial classes that are needed to build more complex expressions, are called *anonymous classes*, and the resulting structure is referred as *graphs*.

OWL Editors

There is a bunch of OWL editors, being the most common the one developed by Stanford University, the *Protègè*[2]. *Protègè* is a Java based application able to edit ontologies from a friendly user interface. Although the performance is not really good and even simple operations take long to be executed, it is balanced with their scalability, as plugins can extend significantly the functionality originally planned. Other alternative editors could have been SWOOP[3], Model Futures OWL Editor[4], Semantic Works[5] or TopBraid[6].

Other ontology language standards

Other alternatives to the OWL standard could have been chosen. Some of the classical computer ontologies achieved in the past great success, but perhaps below from what was expected. They had in its aim being able to collect big areas of knowledge, even the *whole* human knowledge, what indeed is unapproachable. Cyc[7] knowledge database was thus conceived in 1984, and eventually achieved almost 1,000,000 concepts, related with millions of assertions express in the CycL language (a Lisp variant). The other classical alternative would have been the Knowledge Interchange Format (KIF). KIF was created to serve as a syntax for first-order logic being easy for computers to process. It failed to achieve standard category, but enjoyed substantial diffusion.

However all these alternatives left their way to the new Semantic Web generation ontologies. OWL was not born from the scratch, it was the result of the evolution of two other preceding ontology languages oriented to the Semantic Web, specifically DAML[8] (DARPA Agent Markup Language) and OIL[9](Ontology Inference Layer). OIL was born from an European project, it had as a base the RDF(S) syntax, and with a description logic put a strong emphasis on the formal rigor. DAML was American and also departed from RDFS extending it with object oriented features. Their common point,

---

[2]The Protègè Ontology Editor and Knowledge Acquisition System, http://protege.stanford.edu

[3]SWOOP - A Hypermedia Based Featherweight OWL Ontology Editor. http://www.mindswap.org/2004/SWOOP/

[4]The Model Futures OWL Editor. http://www.modelfutures.com/OwlEditor.html

[5]SemanticWorks - visual Semantic Web design tool for RDF and OWL. http://www.altova.com

[6]A Complete Semantic Modeling Toolset http://www.topbraidcomposer.org/

[7]http://www.opencyc.org

[8]http://www.daml.org/

[9]http://www.ontoknowledge.org/oil/

DAML+OIL was the base for the OWL W3C standard. Of course, they are obsolete and OWL supersedes them.

5.2 Working with ontologies

5.2.1 Ontology queries

The information stored in a RDF file can be accessed through queries to a query processor. The query engine, goes a step above XQuery [26] and performs the hard work of accessing the data model.

RDQL

RDQL (RDF Data Query Language [23]) is a query language for RDF designed in 1998. RDQL queries are XML, and superficially resemble that of SQL. For example, the following query retrieves those resources which have the property `requiresAuthorisation`:

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

Although RDQL is widely implemented by RDF frameworks, it is considered obsolete. Instead, SPARQL is emerging as the *de facto* standard

SPARQL

SPARQL (SPARQL Protocol and RDF Query Language [27]) is a newer RDF query language, consisting of triple patterns, conjunctions, disjunctions, and optional patterns.

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:

```
PREFIX rrd:<http://dmag.upf.edu/dmp/CreationModel.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 conceivable setting a server up, one which could answer to SPARQL queries. This could be done by using Joseki[10]. Joseki is an HTTP engine, developed within the W3C consortium, that supports the SPARQL Protocol and the SPARQL RDF Query language.

5.2.2 Ontology access libraries

Representation and access to the representation is well separed. While representation alternatives have been discussed in the previous section, here it will be spoken about access to these representations.

OWL, being a XML, is a neutral technology, and access to the data could come in varied forms. Bat although there are three libraries which offer access to the OWL data, they are actually very similar. The three are Java based and offer approximately the same functionality. We are referring to *Protègè library*, the *OWLAPI* project, and the *Jena* API.

Jena

Jena[11] is the Open Source Java API chosen for this work. The election can be justified as at the time of deciding it was the one which most users had. It is still an active project and the last version (2.5) has been issued in May 2007.

The API promises an easy access to the OWL data but actually it is a really thin layer which provide not much help to the user. The other alternatives are not better and in general these libraries can be disappointing. Jena is designed as an overly complex system, poorly documented, where the actual functionality is gained after a deep knowledge of the underlying

[10] Joseki - A SPARQL Server for Jena. http://www.joseki.org/
[11] Jena http://jena.sourceforge.net

model: it is needed to know the structure of the anonymous classes (described before) and sometimes this is not trivial. Anyway Jena has object classes to represent graphs, resources, properties and literals, and at least they facilitate the tasks of accessing the ontology. Jena can be combined with ARQ, a SPARQL query processor for Jena, and thus answer SPARQL queries, but the answers are also unaware of anonymous classes and graph structure underneath.

5.2.3 Ontology reasoning libraries

Reasoners are formally decoupled from the APIs we have seen before. One of them, though, offers a seamless integration with Jena, and this is called Pellet[12], and for this reason has been chosen as the working reasoning library for our project. It has a MIT license.

In general, reasoners are separed and offer their services through the HTTP protocol. Reasoner come as independent modules because writing a Description Logic reasoner is a non-trival task and because there were some reasoners already done. The way the interface is done, is through a standard protocol, 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.

Apart from the mentioned Pellet that we use, other available reasoners are RacerPro[13] or FaCT++[14]

5.3 Previous ontologies on contract representations

One of the first steps to be given in the development of a computer ontologies consists of reviewing other ontologies to check if they can be reused.

At a first glance, general purpose ontologies hold a huge set of universal terms. These ontologies can be used as upper ontologies, where the new classes to be defined for the developed ontology can be put in relationship with these higher ontologies (*upper ontology* is the ontology which describes general concepts like time or space). Among them, the most preeminent is the SUMO[15] ontology [28], which is regularly maintained and up to date.

[12] Mindswap Pellet, http://pellet.owldl.com/

[13] RacerPro commercial reasoner http://www.racer-systems.com/

[14] FaCT++ GNU Reasoner http://owl.man.ac.uk/factplusplus/

[15] SUMO, Suggested Upper Merged Ontology, owned by IEEE, http://www.ontologyportal.org

As an upper ontology, it could have been used an specific ontology on legal terms as well, for example LODE [29], [30], or LRI-Core [31]. The most interesting and new effort in this sense, is the Estrella European Project[16], which also defines an OWL Ontology of Basic Legal Concepts [32].

Other ontologies are even closer to our domain and so OREL [33], proposes an alternate Rights Expression Language based also on an OWL Ontology aimed at replacing MPEG-21 REL licenses. Some of them emphazise the logical and the inference aspects of ontologies, and so, [34] tries to model the MPEG-21 REL in a CLIPS framework.

Contract ongologies have also existed since the last 10 years, like [35] or [36]. Here the efforts within the DMAG group have been notable, and the IPROnto deserves special mention (see [37] or a shorter summary in [38]). This is an ontology specifically aimed at modeling the Intellectual Property Rights, with several variants to represent related fields (like RDD or REL). IPROnto was proposed for its adoption as MPEG standard [39]; being perhaps too advanced for its time it was rejected. Afterwards in the same group, there were even precedents pointing at an ontological view of the contract representation problem, like [40], but with not much practical work behind.

Experience has shown that overly complex Ontologies are difficult to manage in terms of maintaining both consistency as per available machine based reasoners and corresponding human understanding of the relationships between concepts. These relationships need to be agreed in order for the model to be trusted in its implementation. In this way, a basic set of easily agreed principles can be extended to include any number of scenarios that adhere to the same core set of underlying precepts.

[16]ESTRELLA European project for Standardised Transparent Representations in order to Extend Legal Accessibility IST-2004-027655

Part II

Contribution

Chapter 6

Representation of the value chain as an ontology

The mere fact of expressing a model with a set of logical statements requires a precise analysis that is by itself beneficial for the understanding of the problem. The first result of a modeling process is an abstract knowledge residing in the engineer's mind. This knowledge is implicit. The way this knowledge has been made explicit in this work is through a computer Ontology.

Computer ontologies are the cutting edge in knowledge representation technologies, as has been seen in page 35. Their future, specially RDF and OWL as the core part of the promising Semantic Web, has led the author of this document to decide making it the base for the knowledge representation of agreements. A wide range of ready-to-use applications around OWL is present and boosts development, as well as profits the forthcoming improvements: the formats and tools used along this work are still active, evolving fast and with revised versions being regularly updated. Having chosen OWL as the format for representing the ontology, all these benefits are ensured.

6.1 Methodology

Ontology development is not usually a single event rather a process, and in occasions a fairly complicated process. Because of this, following a methodology in the development of the Ontology is advised.

Ontology development methodology has been long studied, and some good methodologies collections can be found in [41], [42] and [43]. In this work, making a simple ontology was a design objective, and thus the methodology followed has been the simple, following the steps as in [44]. The next list summarizes the tasks which were done when designing the ontology.

- Step 1. Determine the domain and scope of the ontology.

  It has been considered in section 1.2.

- Step 2. Consider reusing existing ontologies

  Existing ontologies such as IPROnto (see Section 5.3) were considered but our ontology was required to keep simplicity at his maximum and a new model was requested. Even this new structure could have been based in a general upper ontology, like SUMO[1], but again the model was decided to remain independent. We believe the approach is valid, as later on more connections can be added on demand.

- Step 3. Enumerate important terms in the ontology

  Informally listed in section 2.2.

- Step 4. Define the classes and the class hierarchy

  This will be done next in section 6.3.1.

- Step 5. Define the properties of classes

  Considered in section 6.3.2.

- Step 6. Define the additional properties related to or necessary for properties (i.e., cardinality, bidirectionality/inverse, etc.)

  Also in section 6.3.2.

- Step 7. Create instances

  This ontology has defined no instances.

- Step 8. Create axioms/rules

  Rules will be seen in the implementations section.

Of course, ontology development is a process rather than a single inspiration work, and the process is iterative, as described in Figure 6.1. The ontology shown in this ontology cannot be considered at all closed, it is instead a captured photo of the state of the ontology up to this date.

[1] SUMO (Suggested Upper Merged Ontology), http://ontology.teknowledge.com

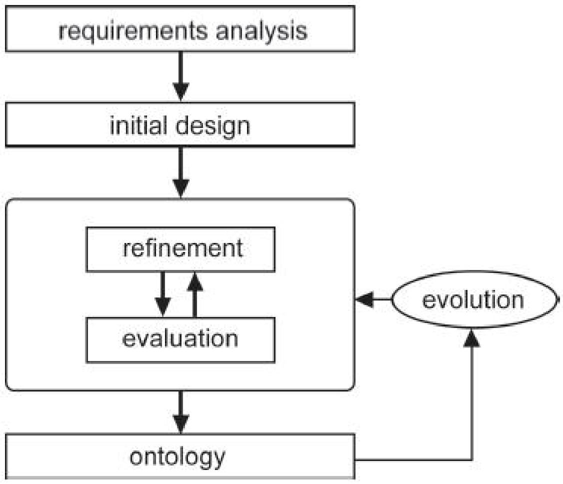


Figure 6.1: Ontology development methodology.

Nr. contracts	*Contract type*	*Subtypes*
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 6.1: Different types found between the analyzed contracts

6.2 Analysis of real contracts

The work done here around contracts has been based on a set of 40 real narrative contracts from the audiovisual market, kindly provided by AFI[2]. These real documents have been provided from several different sources, and represent several different contract types (see Table 6.1), concerning vertically the different steps in the value chain, and horizontally the different media type object of trade (see Table 6.2).

The 40 contracts accounted an average of 8 pages, and 17 clauses each. Although clauses are representative, a single clause sometimes represented several complex ideas, and sometimes just one idea spanned several clauses. Clauses were analyzed and classified, and the following list was extracted as

[2]AFI, Associazione dei Fonografici Italiani, http://www.afi.mi.it/

Nr. contracts	*Material kind*	*Examples*
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 6.2: Different kind of resources traded in analyzed contracts

a summary of the main clauses in agreements on audiovisual material.

Rights The object of the contract is usually the first clause and includes the rights that are being traded.

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.

Territory and Term Restrictions in time and territories appear often as a single clause, and they are very seldom missing in a contract.

Warranties That one party gives to other.

Obligations Additional obligations are not rare in these agreements, specially the obligation of being audited, the obligation of reporting and the obligation of keeping some privacy.

Breach and termination These clauses provision the end of the contract in normal or abnormal conditions.

Jurisdiction In case of dispute, the agreed jurisdiction and court is agreed.

The classification of clauses based on deontic logic, would gather clauses in *claim*, *grant*, *ban*, *duty*. 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.

6.2.1 Contracts and the Value Chain

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 value chain, and we could classify the kind of contracts according to the signing parties. Figure 6.2 shows the typical name of the contract types and relates them with the parties.

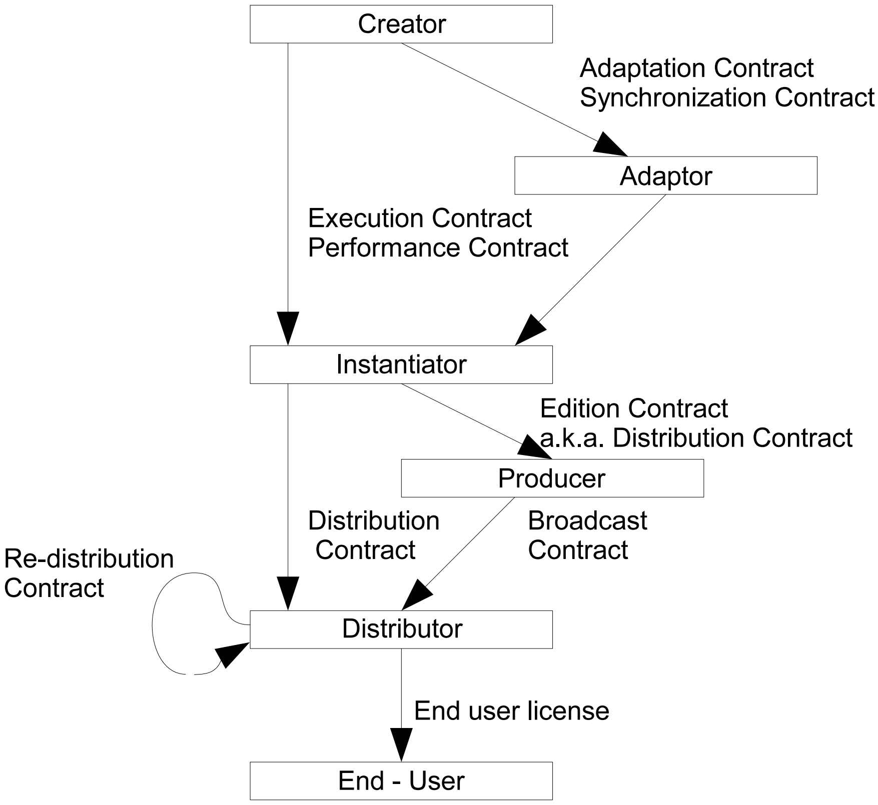


Figure 6.2: Common names of contracts in the Value Chaim.

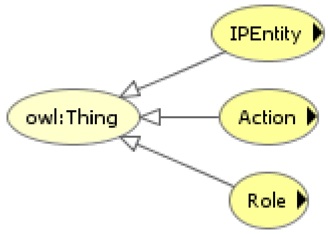


Figure 6.3: Ontology root classes.

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. This is needed to share a common vocabulary. Contracts often have an appendix with the definitions as agreed for both parts, reducing thus ambiguity. Table 6.3 and 6.3 shows only which of these RELs define the term. It is omitted RDD and many other lists of rights could have been used, but this one already gives a good impression about how difficult is agre

6.3 Ontology specification.

The result of the analysis is then expressed as a set of ontology classes, their attributes, the relations between them and eventually some individuals.

6.3.1 Definition of classes

The main three classes are shown in the Figure 6.3. In the OWL file, the definitions of each class are given in the *rdfs:comment* attribute.

Permit, IPEntities, Roles and Actions, are the four top level classes. The three first classes and their basic relationship are shown in the Figure 6.4. The former is *Permit*, used to transfer the capability to perform actions from one user to one or more users.

Users (Creator, Adaptor, Instantiator, Producer...) have Roles associated to them that attribute to them rights over Actions that can be exercised on corresponding IP Entities. Each creation model level is associated with characteristic Actions that correspond with specific functions that relate to activities at different points of the value chain.

	AFI	*DMP*	*REL*	*ODRL*
Reproduce	X			
Perform	X			
Transmit	X			
Broadcast	X	X		
Publish	X	X		
Record	X			
Translate	X			
Dub	X			
Remix	X			
Publicize	X			
Promote	X			
Issue			X	
Obtain			X	
PossessProperty			X	
Revoke		X	X	
Adapt	X	X	X	
Delete		X	X	X
Diminish				
Embed			X	
Enhance				
Enlarge			X	
Execute			X	X
Install			X	X
Modify	X		X	X
Move		X	X	
Play		X	X	X
Print	X		X	X
Reduce			X	
Uninstall			X	X
governedCopy			X	
governedMove			X	
enlist			X	
delist			X	
export			X	
extendRights			X	
governedAdapt			X	
controlledPlay			X	
Extract		X	X	
Access		X		
Adapt Content		X		
Adapt Resource		X		
Distribute	X			

Table 6.3: List of defined rights, part I

	AFI	*DMP*	*REL*	*ODRL*
Backup		X		X
Bundle		X		
Copy	X	X		
CreateWork		X		
Deliver		X		
Edit		X		
Fixate		X		
Grant		X		
Lend		X		X
SubLicense	X			
Move		X		X
Package		X		
Render		X		
Rent		X		
Represent		X		
Restore		X		X
Rewind		X		
PublicCommunication		X		
MechanicalReproduction		X		
Distribution		X		
Store		X		
Stream		X		
Synchronization		X		
Syndicate		X		
Verify		X		X
Withdraw		X		
Display				X
Sell	X			X
Give				X
Lease	X			X
Duplicate				X
Save				X
Excerpt				X
Annotate				X
Aggregate				X
Download	X			
Upload	X			
Make available	X			
Exhibit	X			
Convert	X			
Transcode	X			
License	X			

Table 6.4: List of defined rights, part II

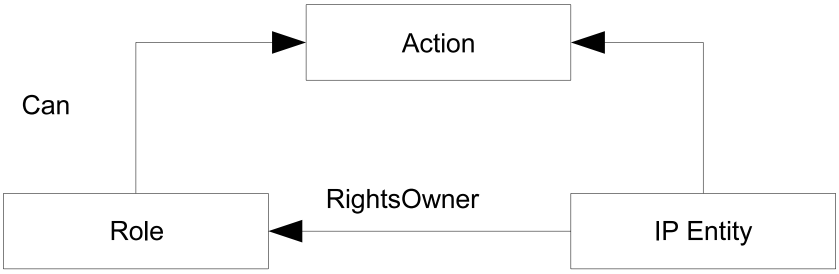


Figure 6.4: Relation among basic classes.

IP Entity classes

An IP Entity is one of the *many identifiable product of the mind attributable to any person(s) or legal entitie(s) that can be represented or communicated physically and protectable by copyright or similar laws.*. The hierarchy of IP Entity classes is shown in Figure 6.5.

IP Entities refer to abstract entities that may be represented digitally such as Work, Adaptation, Manifestation, Instance... Further specialization is given so each can be further refined: For example, Manifestation can be either of an Adaptation (*AdaptationManifestation*) or of a Work (*Work-Manifestation*).

- Work: A creation that retains intellectual or artistic attributes independently of its Manifestations.
- Adaptation: A Work that has been derived from another Work.
- Manifestation: An object or event which is an expression of a Work.
  - AdaptationManifestation: A Content Item that Represents a copy of a Manifestation of an Adaptation of a Work or A Content Item that Represents a Manifestation of an Adaptation of a Work.
  - WorkManifestation: A Content Item that Represents a copy of a Manifestation of a Work or Content Item that Represents a Manifestation of a Work
- Instance: An object or event which is an example of an Identified Manifestation (e.g. a File)
  - WorkInstance: A Content Item that Represents a First Fixation of an Instance of a Manifestation of a Work

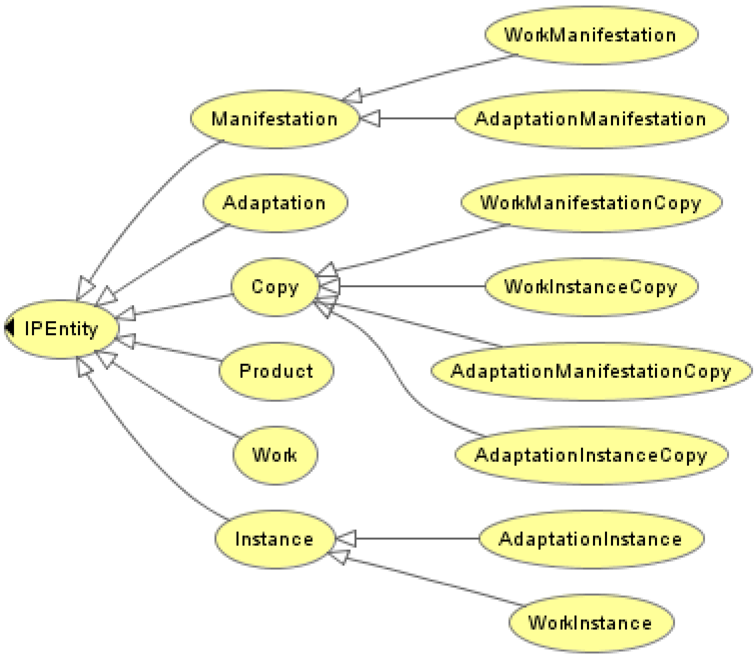


Figure 6.5: Hierarchy of the IP Entity classes.

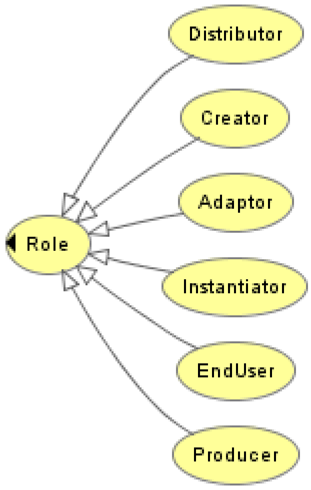


Figure 6.6: Hierarchy of the role classes.

  - AdaptationInstance: A Content Item that Represents a First Fixation of an Instance of a Manifestation of an Adaptation of a Work.

- Copy: The parent class of DMP defined Copies
  - WorkManifestationCopy: A Copy of a WorkManifestation
  - WorkInstanceCopy: A Copy of a WorkInstance
  - AdaptationManifestationCopy: A Copy of an AdaptationManifestation
  - AdaptationInstanceCopy: A Copy of an AdaptationInstance
- Product: A Content Item that adds value to IP Entities by including them with an appropriate Licence for the purpose of Publishing

Role classes

A user is any person or legal entity in a Value-Chain connecting (and including) Creator and End-User. Role is defined as *A defined set of actions and corresponding conditions attributed to and required of a User.*

- Adaptor: A User who produces an Adaptation

- Creator: A User who generates a Work and makes its first Manifestation, also referred to as author
- Distributor: A User who distributes a Product including public communication
- EndUser: A User in a Value-Chain who ultimately consumes Content
- Instantiator: A User who produces an Instance
- Producer: A User who produces a Product from an Instance.

Action classes

An action is simply defined as *the exercise of a right.*

Actions refer to both those that are applied over digital objects and those that are not. The result of some actions may imply the creation of another IPEntity (for example, a MakeAdaptation action generates a new IP Entity of the kind Adaptation) while others do not, for example the action "play". Figure 6.7 depicts the actions represented in the RRD.

- Synchronization. Concurrent performance/display of two distinct Works or Adaptation Instances each for a different sense e.g. text and audio or video and song
- MakeAdaptation: The Right to make an Adaptation
- Produce: The Function of making Products
- Distribute: The Right to sell, rent and lend.
- PublicCommunication: The action of publicly displaying/performing, e.g. live performance, radio, television, internet
  - Stream. The Function of Delivering Content to a Device where the transferred Content is processed for Rendering only and not Stored
  - Broadcast. The Function that Delivers Content to a Device in a point-to-multipoint modality
  - Download. Transfer a file or program from a central computer to a smaller computer or to a computer at a remote location
- CreateWork: The action of creating a Work without any previous material.

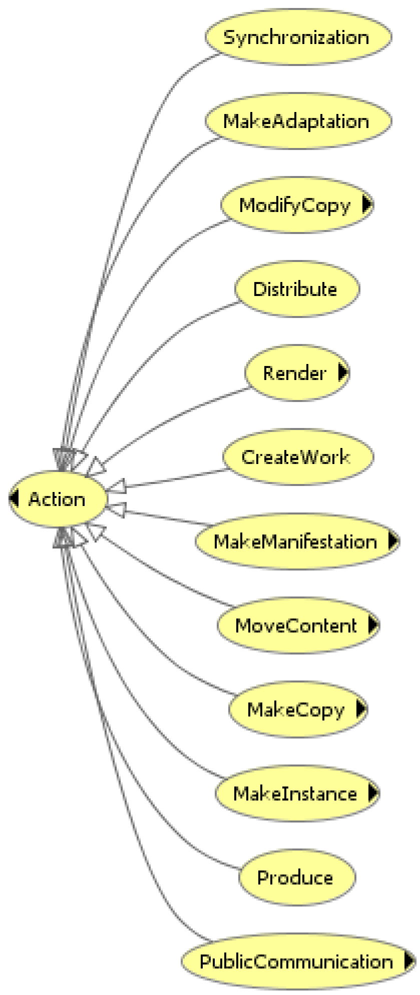


Figure 6.7: Hierarchy of the main action classes.

- MakeCopy. The Function by which Device A Stores Content in Device B, preserving the original Content in Device A. Similar to MechanicalCopy.
- MakeManifestation. The action of making a Manifestation of a Work
- MakeInstance. The action of making an Instance from a Manifestation. Called First Fixation when it is the first time.
- MakeWorkManifestation: The action of making a Manifestation from Work.
- MakeAdaptationManifestation: The action of making an Adaptation Manifestation from an Adaptation.
- MakeWorkInstance: The action of making an Instance from a Work Manifestation.
- MakeAdaptationInstance: The action of making an Instance from an Adaptation
- MakeAdaptationInstanceCopy: Generates a Copy of an Adaptation Instance
- MakeAdaptationManifestationCopy: Generates a Copy of a Manifestation
- MakeWorkInstanceCopy: Generates a Copy of a Work Instance
- MakeWorkManifestationCopy: Generates a Copy of a Work Manifestation
- ModifyCopy. Action of modifying a copy.
  - Enlarge. (MPEG-21 REL)Enlarge represents the Right to modify a resource by making it larger.
  - Reduce. (MPEG-21 REL) Represents the right to modify a resource by taking away from it
  - Extract. (MPEG-21 REL) Extract represents the Right to derive a new resource by taking a fragment out of an existing resource.
  - Embed. (MPEG-21 REL) Embed represents the Right to include a resource in another resource.

  - Export. (MPEG-21 REL) This element represents the right to export the associated broadcast program to another rendering or storage device
  - Modify (MPEG-21 REL) Modify represents the Right to make and save changes to a resource without creating a new resource.
  - Delete. (MPEG-21 REL) Delete represents the right to destroy a digital resource.
  - ExtendRights. (MPEG21 REL) This element represents the right to extend the rights which are the originally transmitted.
  - GovernedAdapt. (MPEG-21 REL) This element represents the right to adapt the resource and results in certain rights being associated with the adapted resource.

- Render. The Function of generating a human-perceivable signal from a Resource

  - Install (MPEG-21 REL) Install represents the right to follow the instructions provided by an installing resource.
  - Uninstall. (MPEG-21 REL) Uninstall represents the right to follow the instructions provided by an uninstalling resource.
  - Print. (MPEG-21 REL) Print refers to the making of a fixed physical representation, such as hard-copy prints of images or text, that may be perceived directly (that is, without any intermediary process) with one or more of the five human senses.
  - Execute. (MPEG-21 REL) Execute represents the right to execute a digital resource.
  - Play. (MPEG-21 REL) Represents the Right to derive a transient and directly perceivable representation of the Resource. The Function of Rendering a Resource

- MoveContent. (DMP-Move) The Function by which Device A Stores Content in Device B deleting the original Content in Device A.

  - Move. (MPEG-21 REL) Represents the right to relocate one resource from one place to another.
  - GovernedMove. (MPEG-21 REL) This element represents the right to copy the resource and at the same time to result in certain rights being associated to the copied resource.

Relation	*Short Definition*	*Func.*	*Domain*	*Range*
Supports	Actions for IP Entity	No	IPEntity	Action
Can	Possible actions for a role	No	Role	Action
RightsOwner	Owner of an IP Entity	No	IPEntity	Role
ResultsIn	Resulting IP Entity	Yes	Action	IPEntity
RightsGivenBy	Which role gives rights	Yes	Action	Role
Origin	Prevenance of an IPEntity	No	IPEntity	IPEntity
HasConsentOver	Special creator consent	No	Role	IPEntity
PermitActions	Allowed actions	No	Permit	Action
PermitIPEntity	Permit IPEntity	No	Permit	IPEntity
SubjectOfPermit	Permit subject	Yes	Permit	Role
ObjectOfPermit	Permit object	No	Permit	Role

Table 6.5: Relations in the ontology

6.3.2 Relations in the ontology

Table 6.5 shows the relations defined in this ontology. Relations bind classes in the Ontology. Every relation has a *domain* corresponding to values from one class resulting in a *range* of values of another or the same class.

A *functional property* is a property that can have only one (unique) value of the range for each value of the domain. When the functional property is absent, it is understood that there may be any number of values corresponding between domain and range.

To represent variations of a range domain correspondence operators such as *min*, *max* and *exactly* can be used. For example, every product *has min 1* copies as sources and a *WorkInstanceCopy exactly 1 WorkInstance Origin*.

To represent variations of a range domain correspondence operators such as *min*, *max* and *exactly* can be used. For example, every product has *min 1* copies as sources and a WorkInstanceCopy *exactly 1* WorkInstance Origin.

The first four relations serve to establish general restrictions. While in the RightsOwner and HasConsentOver do not establish restrictions but illustrate how the RRD can operate with individuals of the Work (MyWork), Creator (Alice) and User (Bob) respectively.

Chapter 7

Implementation

7.1 License Editor

The MPEG group issues the standards but does not provide any implementation excepting a minimalistic reference software. To put the ideas into practice, a minimal license editor software would be the first step to start with.

A minimal license editor would allow creating new licenses, viewing those already existing, and editing them, at least until a simple extent. Actually the MPEG-21 REL license is quite complex and its complete implementation -not yet carried out by anybody- would require a big effort. Therefore, with more humbles objectives in mind, it was designed a C++ application able to cover such minimal functionalities.

The results of the work have been included within the framework of the Integrated Project Axmedis[1] project. This has supposed a real life test, where third persons had to work with the License editor.

License Editor Design

License editor was designed as to perform a basic edition of MPEG-21 Licenses. Editing capabilities include:

- Visualize the license in a tree structure and in panels.
- Store and retrieve licenses from XML files.
- Remote storage of licenses in an authorising server.

[1]Automatic Production of Cross Media Content for Multi Channel Distribution (AXMEDIS), IST 2004 511299, http://www.axmedis.org.

- Possibility of creating distribution or end user licenses.
- Support for the next rights in the end user licenses: *modify, enlarge, reduce, move, adapt, diminish, enhance, embed, play, print, install, uninstall, execute* and *delete.*
- Clustering of rights into grants, which in turn can be grouped into "grantgroups" in a hierarchical structure.
- Support of conditions, which can be added to the grants:
  - *Interval*, to express a temporal restriction (i.e., time when the execution of the rights are allowed). It is followed the Standard ISO 8601, which specifies numeric representations of date and time.
  - *Territory*, to express a spacial restriction (i.e., places where the execution of the rights are allowed). It is followed the standard ISO 3166, which specifies canonic country and area names.
  - *Fee*, to express all the conditions having to do with economic compensations. It can be the amount of money, the currency and the way it will be delivered: at a flat rate, per use, per time etc.
  - *Number*, to limit the number of times a right can be exercised.
- Load and store PARs, (Potential Available Rights), licenses whose issuer and principal have been skipped.
- Search for grants within a license
- Identification of resources and users with URIs. This could be improved in a further version by using specific identifiers for the works, following the ideas of the MI3P initiative [45] (Music-Industry-Integrated Identification-Project).
- Possibility to limit it to a viewer version having its edition functionalities disabled.

Implementation

The MPEG-21 REL license can hold an unlimited number of grants, rights and conditions. Showing a graphical interface able to display an unbounded

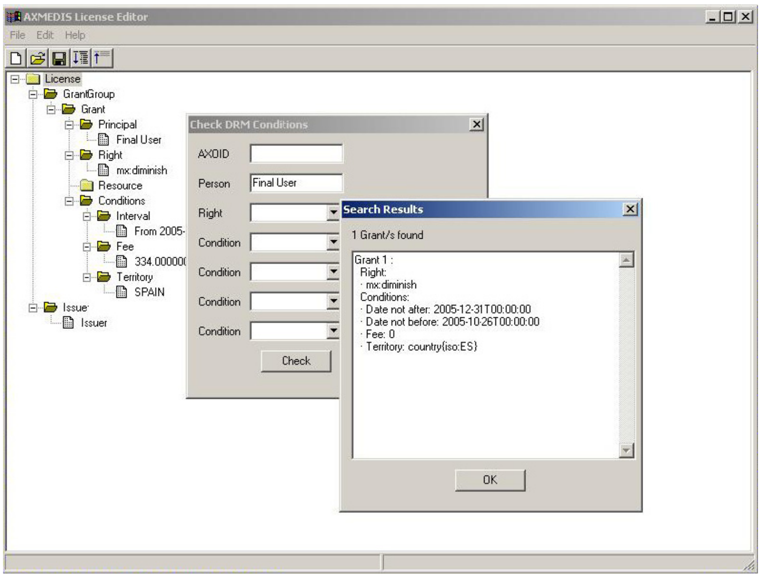


Figure 7.1: Tree view of the DRM Editor.

in size structure, can optimally be accomplished using a tree view (see Figure 7.2). However, this is not the most pleasant way of presenting the information, and thus a limited panel-based version was also created.

The implementation was determined by the context of the project. In particular, parts of the application were based on existing software, such as the modules which manage the license model. In order to comply with this model, the implementation was carried out with the wxWidgets[2] library.

The tool was developed as a library with a user interface that could be plugged to different applications. It was externally presented in two ways, as a standalone application, and integrated in an Axmedis project application (see Figure 7.3).

7.2 Guided Creation of Licenses from Contracts

Previous section has described an application to edit simple MPEG-21 REL licenses. These licenses are oriented either to the end user or to the redistribution of contents (the later being expressed as distributor licenses with the *r:issue* right). If our ambition is to cover the whole value chain of contents

[2] Framework for creating portable applications through Windows, Linux and Mac. http://www.wxwidgets.org/

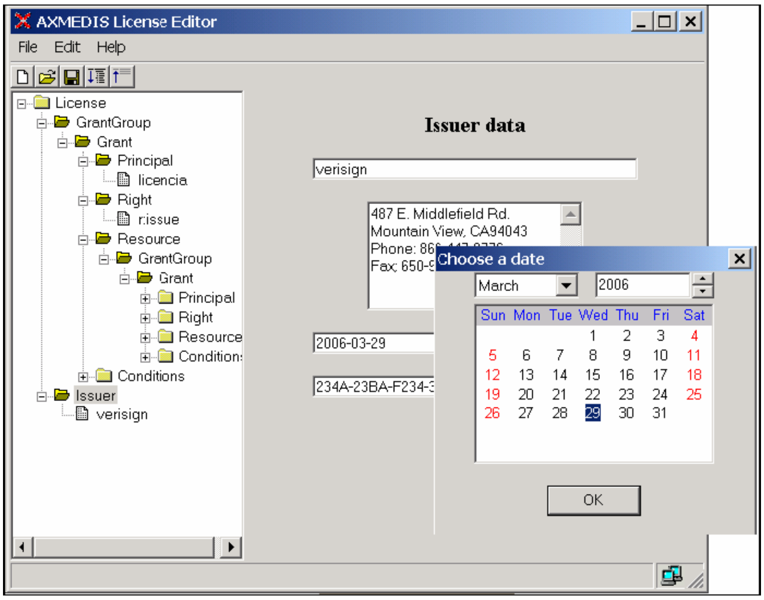


Figure 7.2: View of the DRM Editor.

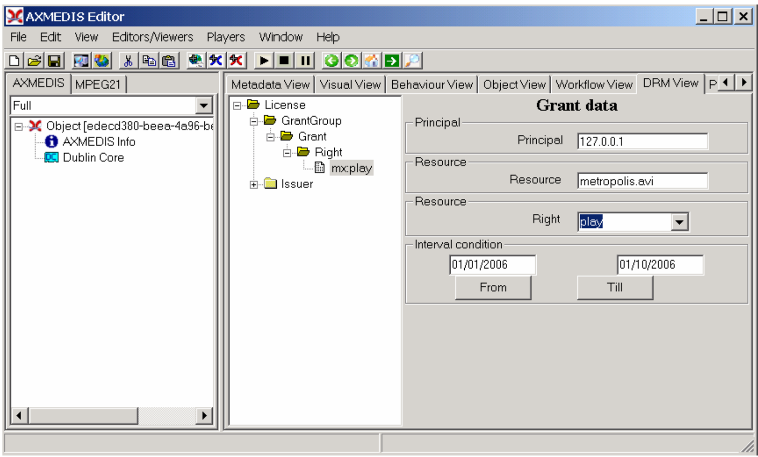


Figure 7.3: Integrated View of the DRM Editor.

Term	*Definition*
Reproduce	To authorize the act of reproduction of content in any manner or form (i.e. reproduction covers all methods of reproduction for instance drawing, lithography, offset and other printing processes, photocopying, recording).
Perform	A performance is considered "public" when the work is performed (presented or executed) 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 and its social acquaintances are gathered. It is more related to audiovisual works.
Transmit	To send data over a communications line
Broadcast	To send out or communicate, especially by radio or television
Publish	To prepare and issue for public distribution or sale
Record	To register (sound or images) in permanent form by mechanical, electrical or electronic means for reproduction
Translate	To render in another language
Dub	To insert a new soundtrack, often a synchronized translation of the original dialog, into (a film).
Remix	To recombine (audio tracks or channels from a recording) to produce a new or modified audio recording
Publicize	To attempt to sell or popularize by advertising or publicity
Synchronize	To cause (soundtrack and action) to match exactly in a film
Digitalize	To convert from analogue to digital form

Table 7.1: New terms added to MPEG-21 REL to support contracts

creation, the MPEG-21 REL license has to be overcome.

Therefore, a new set of terms has been defined and added to that of MPEG-21 REL. The terms definition was carried out jointly with AFI, and it is shown in Table 7.1.

But also the same style of editing has been evolved from the first approach exposed before. This time, the target user of the license editing application would not be intended to be a IT professional, but an occasional user, and thus the editing process has been eased. There are many narrative contracts that are still in force or are model for future contracts. They should be mapped into their corresponding electronic version with the tags described in the previous section, what might be a dull task. The kind of persons involved in this translation may come from a legal background, and the task may be highly repetitive, so having a user-friendly computer guided system to process the contracts would be rather helpful.

An ideal automatic translation system would parse a narrative contract and without human supervision would extract the electronic version from

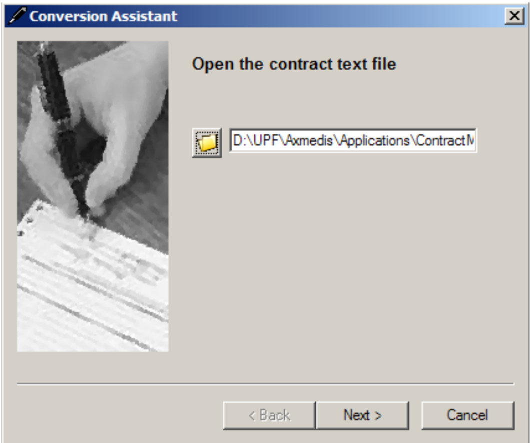


Figure 7.4: Contract Manager Wizard step 1.

it. However, implementing this system is far beyond the state of the art. Instead, what is proposed here is a semi-automated or guided process, where the responsible of the narrative contract can easily extract the electronic version with the support of a computer. So, the user introduces the contract in a text format, follows a computer wizard, and finally obtains a final electronic contract.

7.2.1 Implementation

In its first version, the application was partially implemented as a web application (see [46]), being later transposed as C++ application. The program has been called *Contract Manager* and operates in several steps:

Loading the narrative contract

First, a contract file is loaded (see Figure 7.4). Narrative contract is accepted either as text or in a PDF format.

Parsing the narrative contract

Then, the contract is parsed in an operation hidden to the user. The application converts the text file into an intermediate contract descriptor file. All the sentences in the contracts are statistically analyzed, and those that are likely to belong to one of the given set of clauses, are automatically pre-classified in a new tagged file.

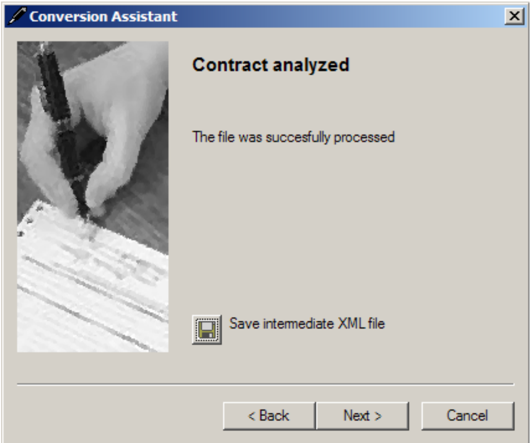


Figure 7.5: Contract Manager Wizard step 2.

This file is structured as an XML file to be the input of the next stage, and could follow an eContracts Schema (see Section 4.2.3). In order to identify these sections and to give a primitive classification of the clauses, a statistical analysis is done.

This analysis bases its decision in a preloaded database, where each of the considered rights and conditions is associated to a set of typical English keywords, keywords that when analyzing the particular contract will be seek. For example, the "territory" clauses, usually include terms such as "country", "territory", "region" or "world" etc. Each of these words receives a ponderation, and when analyzing the text contract, an optimal decision will be taken.

When the analysis is performed, it is shown whether the parsing operation was a succes

```
<?xml version="1.0" encoding="UTF-8" ?>
<contract>
<licensor>Licensor_Company </licensor>
<licensee>Licensee_Company </licensee>
  <term>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 one (1) year
  (\License Period").</term>

  <territory>(c) User intends to acquire, with respect to Archive or part
  thereof, nonexclusive play rights (\Rights") for the territory of Spain
  (\Territory"), subject to the terms and conditions set forth in this
  Agreement.</territory>
```

```
  <fee>10. In consideration of the copyrights materials granted in this
  Agreement and additional fees, User shall pay Distributor a license fee
  of Euro 1 (one) gross, per minute or part thereof used. User states that,
  according to the Italy Convention for the avoidance of double taxation,
  the witholding tax to be applied is equal to the 5% (five per cent) of
  the gross amount of the license fee. Therefore User shall pay Distributor
  a license fee of Euro 1 (one), net. The amount above shall be paid by
  User upon receipt of invoice by check or bank transfer made payable to:
  Licensor_Company</fee>

  <jurisdiction>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.
  </jurisdiction>

  <rights>
    <right>5. User shall use Archive exclusively in a program entitled
    \Program name" 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. </right>
    <right>(c) User intends to acquire, with respect to Archive or part
    thereof, non-exclusive play rights (\Rights") for the territory of Spain
    (\Territory"), subject to the terms and conditions set forth in this
    Agreement.</right>
    <right>(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");</right>
    <right>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.</right>
  </rights>
</contract>
```

Generating the license

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

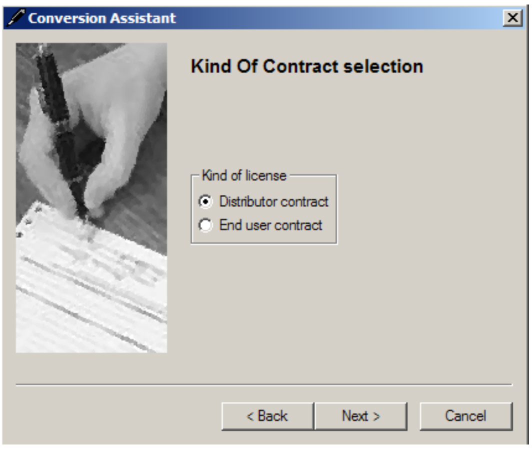


Figure 7.6: Contract Manager Wizard step 3.

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 first one is deciding the kind of contract, for the moment *distribution* contract or *end user* contract. Naturally, there are several others foreseen: *edition*, *performing*, and *production*.

The 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. 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.

The guided process continues asking (Figure 7.7) for the parties in the contract (identified as *Issuer* and *Principal* in the license). Then it will be requested the term, the territory, and the fee conditions. The rights will be asked, and there will be valid some of the REL rights as well as the new rights expressly created for representing the typical REL clauses. Resources as expressed as URIs.

In the last step of the wizard, it will be possible to store the license, to view an equivalent text version of the license created or to send the license to a remote storage service.

For completitude purposes, in the created license, it is kept a version of

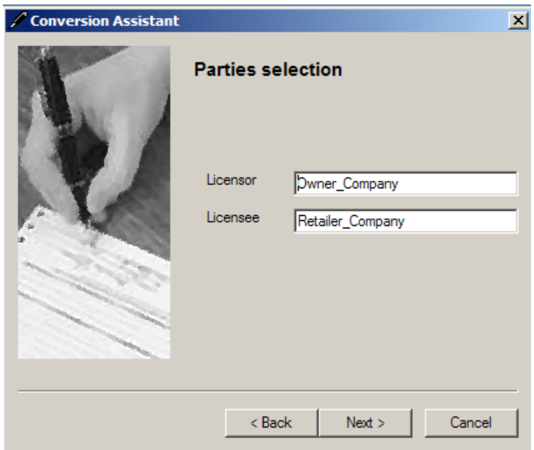


Figure 7.7: Contract Manager Wizard step 4.

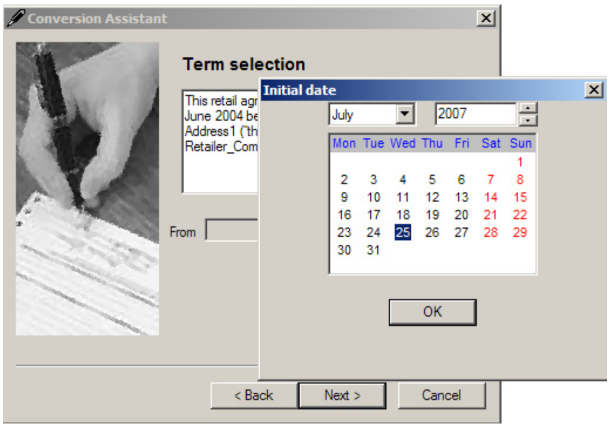


Figure 7.8: Contract Manager Wizard step 5.

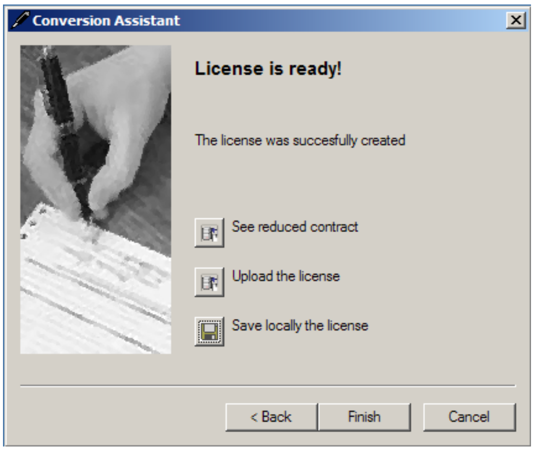


Figure 7.9: Contract Manager Wizard, last step

the literal narrative contract. MPEG-21 REL license defines an *otherInfo* element, intended to store additional content, that can be found appropriate or convenient. The *otherInfo* element can be useful for conveying information which is useful but not part of the REL core infrastructure, and other processors of the license may choose to completely ignore their content. So it has been used to store the complete contract, in a twofold way: as a literal text transcription of the clauses, and as the structured classification of the contracts. The subelement *literalContract* holds a literal copy of the text of the contract, while the element called *contract* holds the clauses classified as it was described in the section 7.2.1.

7.3 Demo: An ontology based application

We have seen in the previous sections the applications developed for creating licenses (either simple REL licenses or REL licenses extended for contracts). The other line of work is developing software based on the semantic representation of the agreements.

7.3.1 Application Programming Interface

The ontology is a XML file and after all applications could work with it directly. However, as it has been seen in Chapter 5, there are libraries facilitating the access to the ontology. But, as it was also commented, these interface libraries lack simplicity and require a previous knowledge of the model. To overcome this and made easier future developments based on this ontology, a specific API has been programmed.

This API is a Java library[3] offering the most common operations over this ontology, so that the underlying layers are hidden and the user of this interface does not need an *a priori* knowledge.

Naturally, the API has been defined as a set of Java interfaces, and it has been provided one implementation too. CreationModelAPI comes under the form of a single .jar file, as it is simple enough as to be contained in a single package. It requires a few other .jar files from its dependent libraries. Three classes are defined in the API:

- CreationModelAPI: Contains the general methods for loading, storing and validating an ontology.
- CreatinModelAPIModel: Contains the general methods for querying general relations.
- CreationModelAPIEx: Contains the methods to deal with individuals.

7.3.2 Demo application

The demo application is based on the API described in the previous paragraphs.

It has a simple user interface, which permits querying about the general characteristics of the Ontology (the three first button in Figure 7.12) and permits the management of a short group of individuals (the fourth button).

The application queries the ontology about the possible actions a role can exercise, by using the API commented in the previous paragraphs (Figure 7.13). Reciprocal to the previous, Figures 7.14 displays the inherent actions associated to the each of the IP Entities. In this case, the relation in the ontology is called "Supports".

Linking together the previous information, this dialog shown in 7.15 (displayed naturally after clicking "Actions") allows the user to choose a "Role" and an "IPEntity". Given this combination, it is shown:

- The possible actions that the selected "Role" can perform over the "IPEntity". They are logically the intersection between what a "Role" can do and what can be done over an "IPEntity".

[3]The general API has been called CreationModelAPI, taking the specific names AxIPOntologyAPI in the context of the Axmedis project, and RRDOnto in the context of the DMP project.

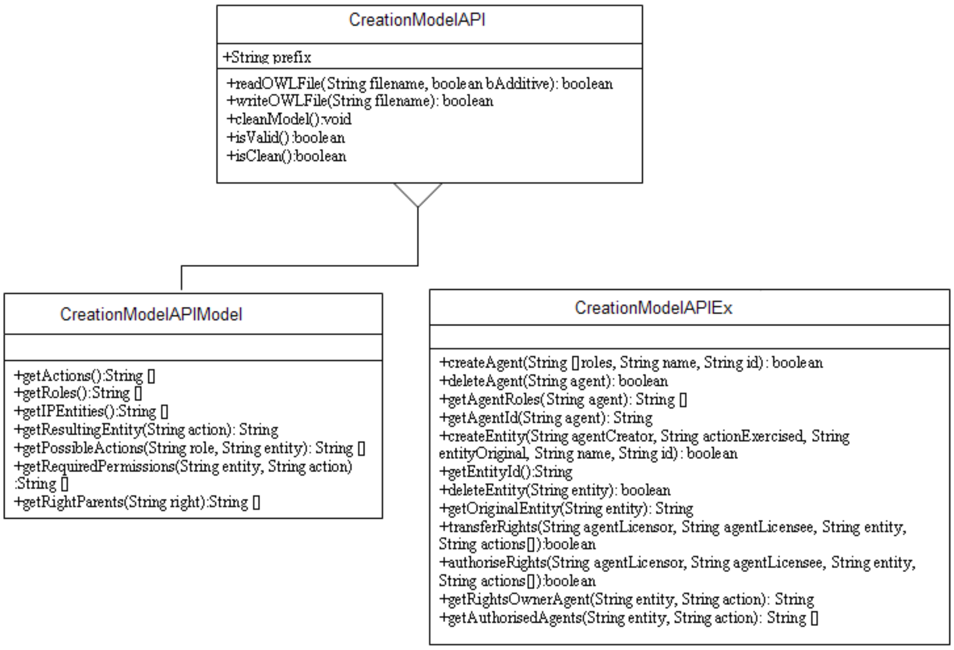


Figure 7.10: Class diagram of the API

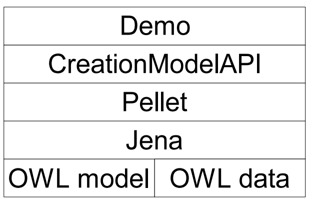


Figure 7.11: Stack of technologies for the demo application

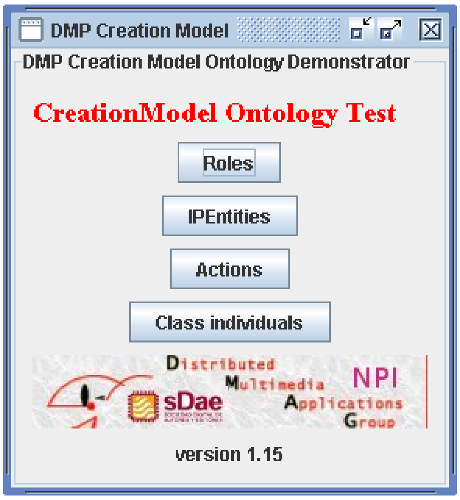


Figure 7.12: Ontology-based demo application.

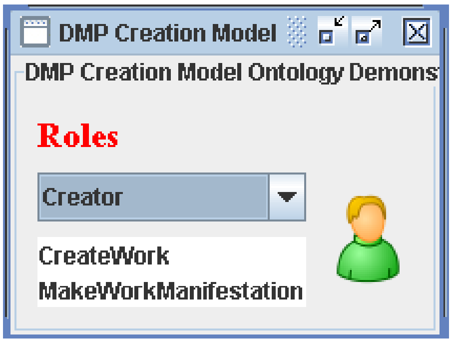


Figure 7.13: Demo application querying roles skills.

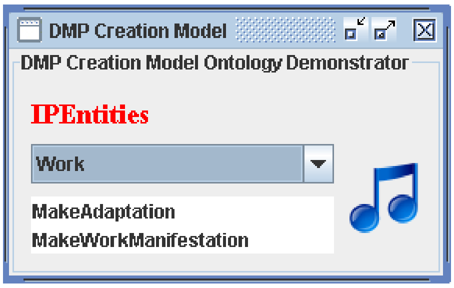


Figure 7.14: Demo application querying on IP Entities.

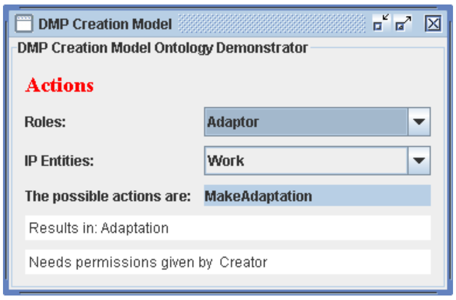


Figure 7.15: Demo application.

- The resulting “IPEntity” that is created as a result of the exercise of the selected “Action” (when more than one actions is possible, then any of them can be chosen by clicking it). Not all the actions generate a new “IPEntity”: doing an “Adaption” action effectively creates a new “IPEntity”, but exercising a “PublicCommunication” does not create a new object. This relation is called in the ontology “ResultsIn”.

- If the role needs a permission to exercise the action. In this case the Adaptor needs permission from the *Creator* to *MakeAdaption*.

The final purpose of the demo application is to show how to deal with individuals. Figure 7.16 displays a panel where individuals can be managed. Users can be created, listed, edited and deleted. Users can execute their rights or they can transmit the rights to other users.

Although it is a simple sample application, it shows how easy can be developing applications based on the API to manage the ontology. In the future, more complete and complex applications should be created.

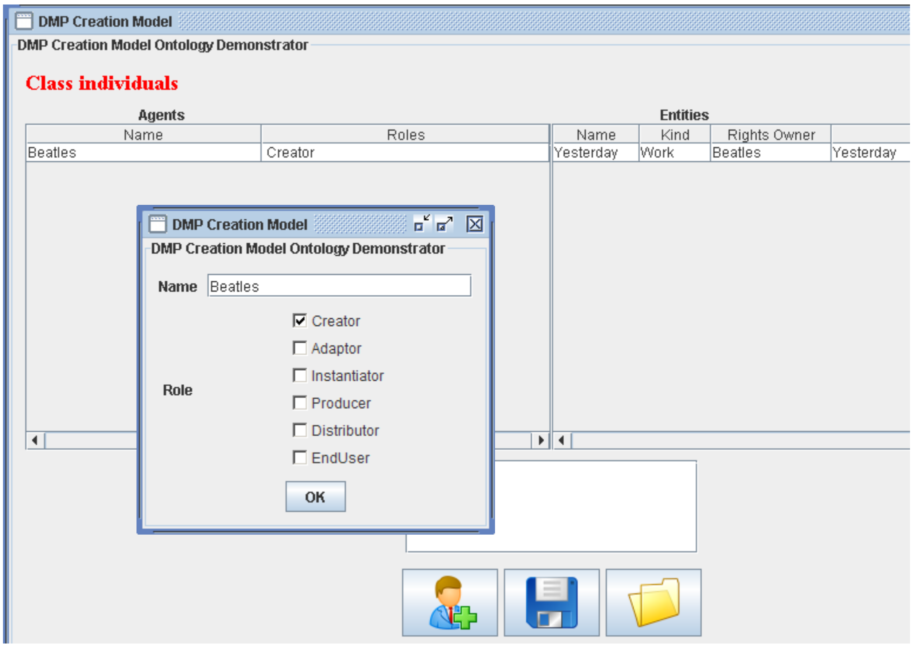


Figure 7.16: Demo application dealing with individuals.

Part III

Future Work

Chapter 8

Conclusions and future work

8.1 Conclusions

A fair acknowledgment of authorship is of interest by itself as long as we have a minimum sense of justice. But it is also the base for a powerful market, the market of the intellectual creations.

While business to consumer business platforms are mature and nowadays massive electronic commerce of digital contents is taking place, business to business transactions still rely massively on the traditional channels and do not take enough advantage of the technology promises.

This is partly due to the existing unclear business models, and this work has been an answer to this demand. Intellectual Property value chain had to be rigorously specified and it had to be done in the most expressive and neutral computer language. OWL was the chosen specific format.

Practical applications can be built based on this OWL representation of the model. Developing such applications is not the focus of this work, but in order to prove its feasibility and its practical use, some simple demonstrators were built.

First, a REL license editor was programmed, able to edit and store REL licenses. These REL licenses proved to be useful as end user authorizations, and as distribution permissions, but were insufficient when attacking other segments in the value chain. These other steps in the value chain are still governed by narrative paper contracts, and REL should be extended to meet this need. With this purpose in mind, the narrative paper contracts were analyzed and its most important information, from a governing utility point of view, was included as a REL extension.

Bringing things into practice, a computer application was created to help the user converting narrative contracts into licenses. This conversion was understood as a semi guided process, where the Artificial Intelligence could assist efficiently the user. Natural language processing is not the object of this work, but its use here could be of great use.

These licenses, either simple REL schemes or RELs extended to support contract terms, have been working at a syntactic level. In parallel, simple demo applications were developed on top of the semantic representation of the OWL. These applications were more modest in scope, and at this stage of development they did not aim at representing full licenses with the conditions and clauses found in the contracts. Instead, they tried to prove how the authorization based on authorship relationships can be built. That is to say, with the ontology as has been described here, it is possible to represent and validate authorisations based on the single criterium of rights property, and its voluntary transmission. The conditions under which this transmission take place would be a further step in the development of the ontology.

However the task has not been completed. First, because this demo applications so far developed, could be taken to its limit of potential. And second, because the ontology must evolve and incorporate terms to represent conditions and the clauses found in contracts.

As a general conclusion, we can state that an incipient work is binding together abstract ideas like ontologies with complex real-life contracts, but for this work to be plenty useful still much work has to be done. The Semantic Web is a cutting edge technology which is conceiving new ideas, and being able to relate these advanced concepts to already existing technologies and systems represents a challenge. It is a challenge following the new trends and at the same time not loosing the perspective of the practical human problems, specially when these problems concern the society so deeply and demand a solution with desperate urgency. Audiovisual contents distribution, or more generally, authored Information distribution is of crucial interest for everybody, because in the Information era, we are virtually all at the same time producers and consumers of information. We hope this work help clarifying a fair attribution of authorship, and we hope the neutrality of computer reasoning help setting a new more fair framework of ideas interchange.

8.2 Work Framework

The work presented so far here has been partially done and founded in the Framework of European projects.

The Axmedis project has backed the initiatives for developing a license editor including the new broader scope that contract derived licenses impose, and within Axmedis has come the priceless help of AFI with its knowledge expertise on contracts in the audiovisual market.

DMP has backed the efforts done in the direction toward achieving a semantic representation of the value chain, and the characterization of the Computer ontology has been done based on their models and supporting knowledge.

From these two different approaches, both projects converge in the higher objectives of covering the whole life of the Intellectual Creations and doing it with a computer ontology. A proposal for standardization has been also carried out in the MPEG group [47].

The work has partially been presented in:

- Presented in:

  Víctor Rodríguez, Marc Gauvin and Jaime Delgado. An Ontology for the Expression of Intellectual Property Entities and Relations, in the *5th International Workshop on Security in Information Systems (WOSIS)*, Funchal, June 2007. [48]

- Accepted as a poster (but not presented) in:

  Víctor Rodríguez, Eva Rodríguez, Silvia Llorente, Jaime Delgado. Ontologies for Expressing Multimedia Content E-Commerce Agreements, in the *1st International Conference on Semantic and Digital Media Technologies (SAMT)*, Athens, December 2006. [49]

- Accepted and to be presented in:

  Víctor Rodríguez, Jaime Delgado and Eva Rodríguez. From Narrative Contracts to Electronic Licenses: A Guided Translation Process for the Case of Audiovisual Content Management, in the *3rd International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution (AXMEDIS)*, Barcelona, November 2007. [50]

8.3 Future Work

This work is about *expression* of agreements on audiovisual resources. So far we have relied the *expression* on an already constructed language, the OWL, and we have tried to span the complete value chain from creator to end user. We have not completed this task but partially and therefore as future work, the first assignment would be finishing this. Currently, our ontology definition lacks the contract terms, and our contract representation lacks the semantic expressivity of the OWL.

This achievement is in sight, but once reached there is still much research to do and a bunch of interesting working lines to follow.

First, once the ontology has reached a stable status (for the moment is still evolving), and has been checked that is technically sound, the API should be revised and optimized. Their functionalities could be served as web services and the RDF could be publicly available under a RDF Server. Second, several case studies should be considered, and it should be seen how can they fit into the ontology model (i.e., in a particular country for a particular media type, identify its roles and IP Objects and try to match to our model).

Third, the way licenses are generated from contracts can be improved. It has been said that Artificial Intelligence can make progress at language processing, and a more clever application could be developed.

Next, we see a problem in how to represent the resources. Of course, the solution can be as easy as an URI, but international identifying standards of intellectual property works should be used. International Standard Recording Code (ISRC) identifying sound recordings and the International Standard Musical Work Code (ISWC) identifying musical works have been in existence since 1989 and 2001 respectively but their deployment is still not ubiquitous and only deal with music catalogs. The initiatives of the MI3P should be followed [45] in a future. As well, the Ontology may include more DublinCore elements, and the software should consider them.

And finally, the abstract problems should be considered, too. OWL is powerful, mature and still with future. But the ambition of every researcher is ever growing and once detected the limits of a technology, it is in his target going further.

OWL is a satisfactory representation of *static* models, but it is a bit awkward when representing dynamic models. This is so in nature, OWL is *Ontology* Web Language, and an ontology, by definition is a description of parmenic beings whose nature does not change. On the contrary, the scenario where our representation of agreements must move, is a dynamic one in perpetual change. Actually, other representation systems such as UML would be more suitable, as they count with dynamic models more efficients than what RuleML would be. Few has been done about reasoning in such dynamic environments, and much could be done. And it would be specially interesting, as this issue has not been dealt in the literature. But working on this would be far beyond the scope of this thesis, where only applied modeling is of interest and not theoretical dissertations. Nevertheless, adopting a few initiatives in this research line could be perhaps profitable at a reasonable cost.

Having all these ideas in mind - more than we can deal with - poses a big challenge for the coming months. And with a Semantic Web still issuing new technologies, and with a dynamic Audiovisual Contents Industry willing to upgrade their business model, it can be taken for granted that new initiatives will arise. The work in the next months promises to be an exciting quest.

Bibliography

[1] The Digital Media Project. Approved Document No. 1, WD 1.0 Technical Reference: Value Chain Functions and Requirements, Version 3.0. Technical report, DMP, July 2007.

[2] Melvin Aron Eisenberg. The emergence of dynamic contract law. *Theoretical Inquiries in Law*, 2(1), January 2001.

[3] Luis Enrique Ocrospoma Pella. Concepto jurídico penal de documento. *Boletín Jurídico Derecho.com*, September 2001.

[4] 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.

[5] WIPO. Intellectual Property Handbook: Policy, Law and Use. Technical Report 489, WIPO Publications, 2004.

[6] The United States Digital Millennium Copyright Act (DCMA), in. http://www.loc.gov/copyright/legislation/hr2281.pdf, 8 October 1998.

[7] European Parliament and European Council. Directive 2001/29/EC of the European Parliament and of the Council on the harmonisation of certain aspects of copyright and related rights in the information society, 22 May 2001.

[8] 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.

[9] Xin Wang, Guillermo Lao, Thomas DeMartini, Hari Reddy, Mai Nguyen, and Edgar Valenzuela. XRML - extensible rights markup lan-

guage. In *XMLSEC '02: Proceedings of the 2002 ACM workshop on XML security*, page 71–79. ACM Press. 69, 2002.

[10] ISO/IEC 21000-5:2004, Information technology — Multimedia framework (MPEG 21) — Part 5: Rights Expression Language, 2004.

[11] ISO/IEC 21000-6:2004, Information technology — Multimedia framework (MPEG-21) — Part 6: Rights Data Dictionary, 2004.

[12] Renato Ianella (ed.). Open Digital Rights Language v.1.1. http://www.w3.org/TR/odrl/, 19 September 2002.

[13] 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.

[14] Stephen Cranefield and Martin Purvis. UML as an ontology modelling language. In *Proceedings of the Workshop on Intelligent Information Integration, 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*, August 1999.

[15] Yao Hua Tan and Thoen W. DocLog: an electronic contract representation language. In *Proceedings. 11th International Workshop on Database and Expert Systems Applications*, September 2000.

[16] Guido Governatori and Zoran Milosevic. A formal analysis of a business contract language. *International Journal of Cooperative Information Systems*, 15(4), 2006.

[17] P.R. Krishna, K. Karlapaplem, and A.R. Dani. *From Contracts to E-Contracts: Modeling and Enactment*, chapter 6, pages 386–387. Springer Sicence, October 2005.

[18] Bernd Schopp, Markus Greunz, and Katarina Stanoevska-Slabeva. Electronic contracting with XML containers. http://www.scientificcommons.org/12748942, 2000.

[19] Content Reference Forum. CEL: Contract Expression Language. (rev. 1). http://www.crforum.org/candidate/CELWG002-celspec.doc.

[20] Cristian Prisacariu and Gerardo Schneider. A formal language for electronic contracts. In Marcello Bonsangue and Einar Broch Johnsen, editors, *9th IFIP International Conference 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.

[21] Comitee Draft 1.0. OASIS eContracts Version 1.0. Technical report, OASIS, December 2006.

[22] M. Ribière and P Charlton. Ontology overview from motorola labs with a comparison of ontology languages. Technical report, Motorola, December, 2000.

[23] Lassila O. and Swick R.R. Resource Description Framework, Syntax Specification. http://www.w3.org/TR/rdf-syntax-grammar, 2004.

[24] Dan Brickley and Ramanathan Guha. RDF Vocabulary Description Language 1.0: RDF Schema, W3C Recommendation. http://www.w3.org/TR/rdf-schema, 10 February 2004.

[25] Peter F. Patel-Schneider, Patrick Hayes, and Ian Horrocks (eds.). OWL Web Ontology Language Semantics and Abstract Syntax. http://www.w3.org/TR/owl-semantics/, February 2004.

[26] Scott Boag et al. Xquery 1.0: An xml query language w3c recommendation. http://www.w3.org/TR/xquery/, January 2007.

[27] E. Prud'hommeaux and A. Seaborne (eds.). SPARQL, SPARQL Protocol and RDF Query Language. http://www.w3.org/TR/rdf-sparql-query/, October 2006.

[28] A. Pease, I. Niles, and J. Li. The suggested upper merged ontology: A large ontology for the semantic web and its applications. In *Ontologies and the SemanticWeb, Papers from the AAAI Workshop. AAAI Press.*, volume 24, 2002.

[29] Takahira Yamahuchi Chizuro Aoki, Masaki Kurematsu. LODE: A legal ontology development environment. In *Proceedings of The Pacific Knowledge Acquisition Workshop (PKAW'96)*, 1996.

[30] John Bagby and Tracy Mullen. Legal ontology of contract formation: Application to ecommerce. In *AAAI Workshop on Contexts and Ontologies*, July 2005.

[31] J. Breuker. Constructing a legal core ontology: Lri-core. In *Workshop on Ontologies and their Applications. Sao Luis, Maranhao, Brazil.*, volume 25, 2004.

[32] University of Amsterdam. OWL ontology of basic legal concepts (LKIF-Core). Technical report, ESTRELLA project (IST-2004-027655), January 2007.

[33] Huiying Li Yuzhong Qu, Xiang Zhang. An ontology-based rights expression language. In *Proceedings of the 13th international World Wide Web. New York, NY, USA*, 2004.

[34] Chun Hui Suen. Efficient design of interpretation of rel license using expert systems. In *Proceedings of the Workshop on Digital Rights Management Impact on Consumer Communications*, January 2007.

[35] Aspassia Daskalopulu and Marek Sergot. The representation of legal contracts. In *Artificial Intelligence and Society*, 1997.

[36] V. Kabilan and P. Johannesson. Semantic representation of contract knowledge using multi-tier contract ontology. In *Proceedings of Semantic Web and Databases workshop, (SWDB 2003)*, 2003.

[37] Roberto García. *A Semantic Web Approach to Digital Rights Management*. PhD thesis, Universitat Pompeu Fabra, November 2005.

[38] IOS Press., editor. *IPROnto: An Ontology for Digital Rights Management. 16th Annual Conference on Legal Knowledge and Information Systems*, 2003. 16th Annual Conference on Legal Knowledge and Information Systems.

[39] Jaime Delgado, Isabel Gallego, Roberto García, Rosa Gil, Enric Peig, and Rubén Tous. Intellectual property rights ontology (IPRONTO). Technical report, Contribution to MPEG-21 RDD-REL CfP M7675, Moving Picture Experts Group, 2001.

[40] OTM Workshops, editor. *Generation of Standardised Rights Expressions from Contracts: An Ontology Approach?*, 1995. LNCS 3762.

[41] Mariano Fernández. Overview of the methodologies for building ontologies. In *Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods (KRR5)*, August 1999.

[42] D. Jones, T. Bench-Capon, and P. Visser. Methodologies for ontology development. In *Procceedings of the IT and KNOWS Conference, XV IFIP World Computer Congress*, August 1998.

[43] Helena Sofia Pinto and João P. Martins. Ontologies: How can they be built? *Knowl. Inf. Syst.*, 6(4):441–464, 2004.

[44] 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.

[45] RightsCom. The MI3P framework: Introductory workshop proposal. Technical report, RightsCom Limited, January 2005.

[46] Joan Vendrell Farreny. Formalització de contractes i llicències en aplicacions de distribució de continguts multimèdia, September 2006.

[47] 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.

[48] 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.

[49] Víctor Rodríguez, Eva Rodríguez, Silvia Llorente, and Jaime Delgado. Ontologies for expressing multimedia content e-commerce agreements. In *1st International Conference on Semantic and Digital Media Technologies*, December 2006.

[50] Víctor Rodríguez, Jaime Delgado, and Eva Rodríguez. From narrative contracts to electronic licenses: A guided translation process for the case of audiovisual content management. In *3rd International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution*, November 2007.

Typesetting made with LaTeX. All the figures shown were prepared with Open Source Software (Argos UML and OpenOffice), and those external images were reproduced with permission. Barcelona, July 2007.