A Semantic Web approach to Digital Rights Management

by

Roberto García González

Ph.D. Thesis

Doctorate in Computer Science and Digital Communication
Department of Technologies

Advisor: Dr. Jaime Delgado Mercé

Universitat Pompeu Fabra

Barcelona, November 2005

Dipòsit legal: B.26401-2007
ISBN: 978-84-690-7821-1

A Rosa

Acknowledgments

Gracias Rosa, sin ti esta tesis no hubiese visto la luz.

Gracias a mis padres, hermana, abuelos, tíos... siempre me habéis motivado a seguir. Recién acabada la presentación del proyecto final de carrera ya me preguntabais por lo que vendría después.

Gracias a mi director de tesis, Jaime Delgado, sobretodo por aquella recomendación a finales de 1999 sobre que me mirase algo llamado RDF... ahí empezó todo.

I would also like to acknowledge the following people. Aldo Gangemi, Jos Lehmann, Carola Catenacci, Massimiliano Ciaramita, Daniela Tiscornia and Maria Teresa Sagri for their feedback during the Legal Modelling Seminar held in Rome on June 2005. John F. Sowa for his encouraging comment about the "State of the Art" part. And Adam Pease for his comments about the mapping to SUMO.

Abstract

In order to improve the management of copyright in the Internet, known as Digital Rights Management, there is the need for a shared language for copyright representation. Current approaches are based on purely syntactic solutions, i.e. a grammar that defines a rights expression language. These languages are difficult to put into practice due to the lack of explicit semantics that facilitate its implementation. Moreover, they are simple from the legal point of view because they are intended just to model the usage licenses granted by content providers to end-users. Thus, they ignore the copyright framework that lies behind and the whole value chain from creators to end-users.

The contribution of this work is to apply a semantic approach based on web ontologies to Digital Rights Management. The main contribution is the development of a copyright ontology that puts this approach into practice. It models the copyright core concepts for creations, rights and the basic kinds of actions that operate on them. Altogether allows building a copyright framework for the complete value chain. The set of actions operating on content are the building blocks that combined cope with the complexity of the copyright domain. At the same time, their simplicity guarantees a high level of interoperability and evolvability. The resulting copyright modelling framework is flexible and complete enough to model many copyright scenarios, not just those related to the economic exploitation of content.

Additionally, the ontology design and the selection of tools result in a straightforward implementation. Rights are modelled as classes of actions, action patterns are modelled also as classes and concrete actions are modelled as instances. Then, to check if some right or license grants an action is reduced to check for class subsumption and instance classification, which are the main functionalities of Description Logic reasoners. These checks are guided by the modal operators implicit in some of the case roles used in the ontology.

An additional contribution is to apply the same approach to the main rights expression languages, which are based on syntactic solutions. For each of these initiatives, a web ontology has been developed that captures the language grammar but also formalises its implicit semantics. Thus, it is easier to develop tools for these languages and they can be integrated in the general framework of the Copyright Ontology. The integration produces benefits in both directions. On one hand, the copyright ontology can benefit from it because new requirements are detected an it can be evaluated against real world needs. On the other hand, the copyright ontology can contribute its formal semantics to these syntax-based initiatives.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The objective of this work is to make a new contribution to the Digital Rights Management (DRM) research field. There are different initiatives trying to solve the problem of interoperability between DRM Systems (DRMS), which have started from isolated and proprietary initiatives. However, they are lately clearly moving to a web-broad application domain.

One of the main initiatives is MPEG-21 [22], an ISO/IEC standardisation framework for digital content management. MPEG's DRM modelling part is divided into the Rights Expression Language (REL) and the Rights Data Dictionary (RDD) [120]. Another initiative is ODRL (Open Digital Rights Language), available also as W3C note [64], that has been adopted by the Open Mobile Alliance (OMA[1]) as a standard for the mobile communications field.

There are many other initiatives but, basically, all have one thing in common, they work at the syntactic level. Their approach is to define some XML Schemas that specify the grammar of rights expression languages (REL). In some cases, the semantics of these languages, i.e. the meaning of the expressions, are also provided but formalised separately as rights data dictionaries (RDD). Rights dictionaries list terms definitions in natural language, solely for human consumption and not easily automatable.

However, the syntactic approach does not scale well in really wide and open domains like the Internet. Automatic processing of huge amounts of metadata coming from many different sources requires machine understandable semantics. The syntax is not enough when unforeseen expressions are met. Here is where semantics come to help their interpretation to achieve interoperation.

There are other initiatives that have also chosen a semantic approach for DRM. The Harmony

[1] http://www.openmobilealliance.org

project [63] integrates copyright concepts from the MPEG-21 RDD into a generic ontological framework and OREL [94] is also a formal ontology version of MPEG-21 RDD. However, these initiatives do no take into account the copyright legal framework, as the DRM initiatives they are based on do not consider this aspect either. On the other hand, there is the Creative Commons initiative [78], which is also based on semantic metadata but it does consider the legal framework. In this case, the inconvenient is that it provides a very simple formalisation intended for open release environments, e.g. open source software.

The proposed approach is to facilitate the automation and interoperability of DRMS integrating both parts, the Rights Expression Language and the Rights Data Dictionary. This objective can be accomplished using ontologies, which provide the required definitions of the rights expression language terms in a machine-readable form. Thus, from the automatic processing point of view, a more complete vision of the application domain is available and more sophisticated processing can be carried out.

The selected ontologies are those from the Semantic Web approach [13] because they are naturally prepared for the Web domain, they are called web ontologies [56]. The modularity of web ontologies, constituted by concept and relation definitions openly referenceable as URIs, allows their easy extension and adaptation to meet evolvability and interoperability.

The contributed web ontology that formalises copyright law aspects is based on the World Intellectual Property Organisation (WIPO[2]) recommendations, which try to define a common worldwide legal framework. Using a so general framework helps building a general copyright ontology, which can be then concretised for specific law systems. In any case, the current tendency is to adapt local copyright systems to this international framework. This facilitates the interrelations among these legal systems as they are being forced to interoperate due to copyright globalisation.

A preliminary version of this ontology, called IPROnto (Intellectual Property Rights ONTOlogy[3]), was contributed [27] to MPEG-21 REL-RDD call for proposals [82].

1.1 Outline

This document is organised as follows: after the introduction, the state of the art section analyses the four pillars on top of which this research work has been carried out. First, there is the Knowledge Representation chapter. Its content is grounded in a definition of knowledge and then moves to how knowledge is formalised with Knowledge Representation techniques. Nowadays,

[2] http://www.wipo.org
[3] http://dmag.upf.es/ontologies/ipronto

they have been combined with computation resources so this first state of the art section ends with an analysis of knowledge representation technologies.

Second, the state of the art focus moves to the Web. A short description is done centred on its foundation technologies. The main attention is placed on some actual web scenarios that highlight some of its problems. Then, some recent attempts from the Web community to reduce these problems are presented. They are a set of technologies based on the Web Services model.

Then, the third chapter of the state of the art explains the Semantic Web initiative as a combination of the previous Web and Knowledge Representation ideas. The Semantic Web is presented as an attempt to complete the work initiated by the Web initiative in order to overcome its problems. The complete set of design principles is shown and the architecture were they are been realised is detailed. Finally, the Web Services model is reconsidered complemented by the Semantic Web perspective.

The state of the art finishes with a chapter devoted to rights expression languages (RELs), which are the representational part of digital rights management systems. RELs are mainly based on syntactic approaches, i.e. a grammar defines the language elements. Their focus is on modelling the kind of licenses established by media distributors and end-users, which grant them permission for concrete actions under some conditions and constraints. Therefore, they do not deal with the underlying copyright law framework.

Once the research domain picture has been built, the objectives and the methodology are described in the preparatory part. They guide the research work that has been done in the contribution part. This part documents the development process of the main contribution of this work, the copyright ontology, as a conceptual and computable model.

It also includes an evaluation chapter with additional contributions geared towards demonstrating the benefits of the Semantic Web approach to the other REL initiatives and how their integration with the copyright ontology helps checking its validity.

Finally, the conclusions and the future work are presented. The conclusions section reviews the objectives and hypothesis in order to show how they have been fulfilled and confirmed during the contribution part of the work. Moreover, the conclusions detail all refereed publications and contributions to DRM standards that have allowed discussing these contributions in relevant international forums.

Part I

State of the Art

Chapter 2

Knowledge Representation

The best option to understand what Knowledge Representation is simply to mention what it is intended for. Its mission is to make knowledge as explicit as possible. This is necessary because knowledge is stored in implicit form, i.e. tacit knowledge non-observable from the outside, inside minds and spread around in community social habits. To facilitate knowledge sharing it is necessary to make it explicit.

2.1 Tacit Knowledge

Tacit knowledge is what an agent obtains when it observes its environment and makes internal representations of what it perceives. Here "agent" stands for an entity capable of election. Agent choices are built from its internal representation, its model of the world. The model captures what there is and how it works, thus allowing the agent to predict what would happen if it does something or not, a complete view on that from a Systems Theory perspective [72] is shown in Figure 2.1.

In other words, tacit knowledge allows an agent to choose the best options that, hopefully, will help it achieve its goals. These goals are unimportant from a generic point of view. They might range from survival to booking a ticket, passing through getting a favourable transoceanic export rate, for instance.

For social agents, tacit knowledge is also stored distributed in common habits established in a community [93,110]. The same principles apply, although from the perspective of the whole community as an agent. It can be also considered tacit because it is not explicitly represented in

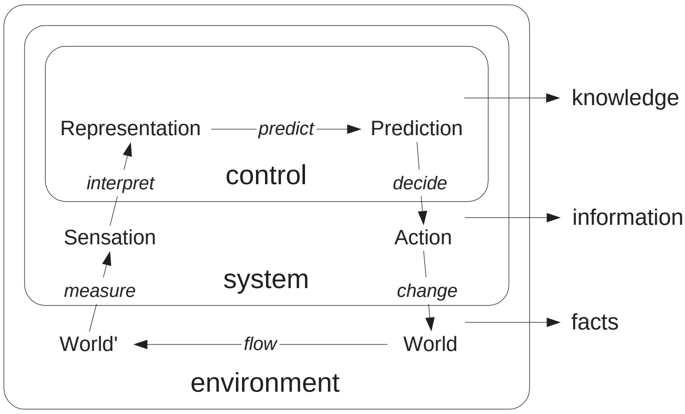

Figure 2.1: Knowledge viewed from Systems Theory perspective

the community. It is distributed while agent act collectively, for example by imitation. This process is also known as socialisation [85], a complete view of the tacit-explicit knowledge cycle is shown in Figure 2.2. Human natural languages are an example of tacit shared knowledge. Although a part of natural languages can be formalised, humans acquire natural language abilities mainly by imitation.

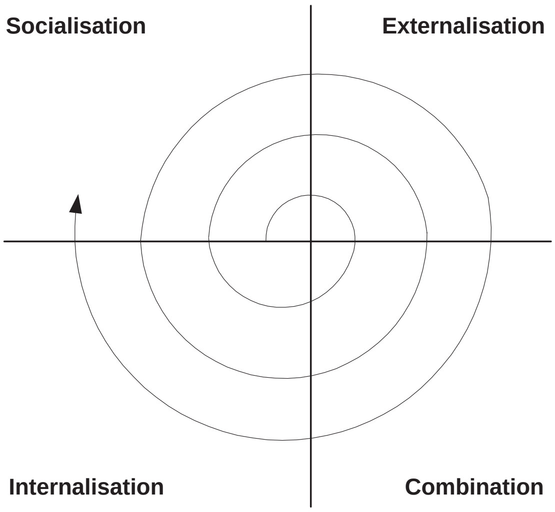

Figure 2.2: The Knowledge Spiral

2.2 Knowledge Sharing

Knowledge is exchanged between social agents because this way each agent gets access to more than the knowledge it has been able to build up. Obviously, each agent, and the community as a whole, is then more prepared to make the correct choices. Agents have access to more than individual experiences and even unprecedented situations can be resolved satisfactorily.

However, tacit social knowledge is exchanged inefficiently. The exchange mechanisms, e.g. imitation, are restricted to local range. This reduces knowledge propagation in space and time.

To overcome these limitations, some agents have developed ways to make knowledge explicit and encode it in more perdurable form. Human languages are an example of this. They have written forms as perdurable encoding, always with some kind of physical support. Moreover, technology advances have also allowed perdurable encoding of languages oral form.

Other kinds of agents have developed external and perdurable knowledge formats. For instance, cells DNA can be considered an encoding of how to reproduce a cell thus allowing its perpetuation.

Generally, all these knowledge transmission mechanisms are studied by Semiotics. They are based on signs, their basic components. Signs stand for things in the agents' domain. They are the building blocks of agents' internal world models and they are encoded and transmitted during knowledge sharing. More details about Semiotics are shown in Figure 2.3 and the Semiotics section 2.7.1.

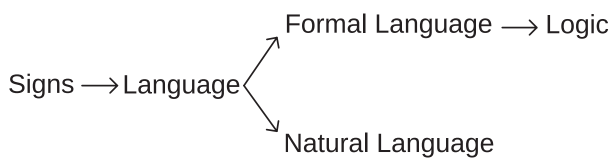

Figure 2.3: Knowledge exchange mechanisms based on signs and thus studied by Semiotics

As has been said, explicit knowledge can overcome space-time limitations of tacit knowledge exchange. Perdurable encoding and transmission mechanisms allow that it can be acquired a long time after its encoding and far away its origination point.

However, many of these representations carry interpretation ambiguities. This is because they are not wholly formalised. They are so expressive that some exchanged knowledge can be acquired at the destination leading to a different piece of knowledge.

However, this cannot be considered a bad property. Ambiguity provides easy adaptation of the representation mechanisms to new situations. For instance, metaphor produces a new interpretation of a previous representation inside a particular context. Ambiguity also allows exploration of new possibilities because knowledge is not confined in a restrictive immutable form. A good example of this advantage is DNA.

2.3 Information

The previous risk of misunderstanding during knowledge exchange is the reason why this kind of encoded "knowledge" should not be considered knowledge. It is more appropriate to say that it is information. The encoded knowledge can be completely lost if the receiver agent cannot understand it. For instance, if two agents exchange a written message but the second one does not understand the used language, nothing of the originally codified knowledge can be retrieved.

Therefore, to be completely strict, there does not exist more knowledge than tacit knowledge. Information is the small part of it an agent is able to articulate [108]. When an agent receives some information, it uses its tacit knowledge to interpret it and, possibly, this may lead to a change in the tacit knowledge it possesses.

However, this categorisation is in practice relaxed. There are different kinds of information and normally when the exchanged information is rich enough it is considered knowledge. Rich information has embedded enough contextual information to facilitate its full interpretation. Moreover, some encoding restrictions must be imposed in order to guarantee, to some extent, a final interpretation near to the original encoded knowledge.

In the opposite side of information rich enough to be considered knowledge, there is data. It is de-contextualised information, i.e. too distant from the knowledge required to interpret it.

2.4 Knowledge Formalisation

As mentioned before, despite ambiguity advantages, sometimes it is necessary to exchange knowledge as reliably as possible. This has been a clear requirement in human societies for a long time. Indeed, Socrates can be considered a starting point in this formal knowledge exchange research, but roots could be extended even before.

From these remote times, humans have developed many representation formalisms. All them define their own set of shared constraints that must be incorporated as tacit knowledge in knowledge emitters and receiver. Once a formalism has been incorporated in the tacit knowledge of a community, this community can share information in a so direct and rich way that it can be considered knowledge exchange.

These formalisms can be very simple, for instance defining a set of reserved natural language words with an agreed community meaning. Then, community agents can share knowledge interchanging messages that use these agreed words. This is an example of a purely textual formalism, but there are also graphical ones. They are called diagrammatic formalisms and they are quite simple and easier to interpret, for instance Conceptual Maps [86].

However, the more powerful formalisms use techniques that are more sophisticated. They are mainly based on mathematics, philosophy and cognitive science. These disciplines provide basic ideas of how we perceive and model the world. Thus, they set a base that we naturally share, although not in an obvious way.

Mathematics provides a compact set of principles widely shared among human society. This shared common base allows the construction of very powerful expressions. These expressions have clear meaning for those that incorporate the used part of the shared mathematical base into their tacit knowledge.

Meanwhile, philosophy studies the nature of knowledge, how we create and manage it. Some techniques have been developed that capture a part of our brain operation. Most of them use mathematical tools to some extent. For instance, logic and ontology are two building blocks of Knowledge Representation. On the other hand, there are also attempts to explain mathematics from a philosophical point of view [79].

Despite all the possibilities of advanced representation formalism, it is important to remark that tacit knowledge is richer than any description of it.

2.5 Knowledge Representations

As has been shown along the previous sections, the final objective of knowledge representations is to make knowledge explicit. Knowledge can be shared less ambiguously in its explicit form and this became especially important when machines started to be applied to facilitate knowledge management.

Nowadays, Knowledge Representation is a multidisciplinary field that applies theories and techniques from:

- **Logic**: provides the formal structure and rules of inference, more details in the Logic section 4.8.7.
- **Ontology**: defines the kinds of things that exist in the application domain, see the Ontology section 4.8.5.
- **Computation**: supports the applications that distinguish knowledge representation from pure philosophy.

Therefore, Knowledge Representation can be defined as the application of logic and ontology to the task of constructing computable models of some domain [106]. Logic and Ontology provide the formalisation mechanisms required to make expressive models easily sharable and computer aware. Finally, thanks to computational resources, great quantities of knowledge expressed this way can be automated. Thus, the full potential of knowledge accumulations can be exploited. However, computers play only the role of powerful processors of more or less rich information sources. The final interpretation of the results is carried out by the agents that motivate this processing, in this case human users of the knowledge management systems.

At this point, it is important to remark that the possibilities of the application of actual Knowledge Representation techniques are enormous. Knowledge is always more than the sum of its parts and Knowledge Representation provides the tools needed to manage accumulations of knowledge and the World Wide Web is becoming the biggest accumulation of knowledge ever faced by humanity. These possibilities will be more deeply explored in the next State of the Art sections, devoted to Web Technologies 3 and the Semantic Web 4.

2.5.1 Principles

In addition to the previous definition, Knowledge Representation can be also described by the five fundamental roles that it plays in artificial intelligence; they are the Knowledge Representation principles [95]:

- A knowledge representation is a **surrogate**: symbols are used to represent external things that cannot be stored in a computer, i.e. physical objects, events, and relationships. Symbols are surrogates for the external things. Symbols and links between them form a model of the external system that can be manipulated to simulate it or reason about it.
- A knowledge representation is a set of **ontological commitments**: Ontology is the study of existence. Thus, ontology determines the categories of things that exist or may exist in an

application domain. Those categories set the ontological commitments of the application designer or knowledge engineer.

- A knowledge representation is a **fragmentary theory of intelligent reasoning**: to support reasoning about modelled things in a domain, a knowledge representation must describe their behaviour and interactions. The description constitutes a theory of the application domain. It can be stated, for instance, as explicit axioms or compiled into computable programs.

- A knowledge representation is a **medium for efficient computation**: besides representing knowledge, an Artificial Intelligence System must encode knowledge in a form that can be processed efficiently by the available computing equipment. Therefore, developments in computer hardware and programming theory have a great influence on knowledge representation.

- A knowledge representation is a **medium for human expression**: a good knowledge representation language should facilitate communication between the knowledge engineers who manage knowledge tools and the domain experts who understand the application domain. Domain experts should be able to read and verify the domain definitions and rules written by knowledge engineers.

2.5.2 Levels of representation

When applied in the computer domain, knowledge representations range from computer-oriented forms to conceptual ones nearer to those present in our internal world models. Five knowledge levels can be established using this criterion [15]:

- Implementational: this is the more computer aware level. It includes data structures such as atoms, pointers, lists and other programming notations.

- Logical: symbolic logic is inside this level. Thus, symbolic logic propositions, predicates, variables, quantifiers and Boolean operations are included.

- Epistemological: a level for defining concept types with subtypes, inheritance, and structuring relations.

- Conceptual: the level of semantic relations, linguistic roles, objects and actions.

- Linguistic: the more computers distant level, it deals with arbitrary concepts, words and expressions of natural languages.

2.6 Logic

This is one of the fundamental aspects of knowledge representation as presented in the Knowledge Representations section 2.5. Logic was developed as an attempt to create a universal language based on mathematical principles. Therefore, it is based on formal principles that impose some requirements over a knowledge representation language to be a logic:

- **Vocabulary** it is a collection of symbols represented as chars, words, icons, or even sounds. These symbols are divided in four groups:
  - Logical symbols: they are domain-independent, e.g. quantifiers like "$\forall$" or connectives like "$\wedge$".
  - Constants: these are domain dependent and identify individuals, properties or relations in the application domain, or universe of discourse. E.g. “truck281” or “motherOf”.
  - Variables: they are unbounded symbols whose range of application is governed by quantifiers.
  - Punctuation: these are utility symbols that separate or group other symbols, e.g. commas and parenthesis.
- **Syntax**: a logic must have grammar rules that determine how symbols combine to form well-formed sentences.
- **Semantics**: it is necessary to make meaningful statements. It comprises a theory of reference that determines how the constants and variables relate to things in the universe of discourse. Moreover, it also includes a theory of truth to distinguish true statements from false. More details about semantics are presented in the Semantics section 2.7.
- **Inference**: this aspect is important in order to get something more than a notation. Inference is carried out by rules that determine how patterns are generated from others. Appropriate inference rules allow reasoning mechanisms automation and, thus, the generation of new knowledge from previous one. This point is detailed in the Inference section 2.8.

Natural Languages can represent a wider range of knowledge, however, logic enables the precisely formulated subset to be expressed in computable form. On the other hand, although there are some kinds of knowledge not expressible in logic, such knowledge cannot be represented either on any digital computer in any other notation. The expressible power of logic includes every kind of information storable or programmed on any digital computer.

2.6.1 Logic types

There are many logic types; each one particularly suited to its target application domains and with despair expressive capabilities and computational requirements. Logic systems vary along six dimensions from what can be considered the reference logic, First Order Logic (FOL):

1. **Syntax**: the most obvious but least important difference among logics is notation. In terms of expressive power, syntactic differences are unimportant.
   - *Typed FOL* is a syntactic extension of FOL, it has identical semantics and there exist direct syntactical substitutions to translate between them:
     $(\forall x{:}t)\phi(x) \equiv (\forall x)(t(x) \rightarrow \phi(x))$ and $(\exists x{:}t)\phi(x) \equiv (\exists x)(t(x) \wedge \phi(x))$.
2. **Operators**: each logic defines a set of permissible operators or combinations of them.
   - *FOL* has the common Boolean operators: conjunction ($\wedge$), disjunction ($\vee$), negation ($\neg$), implication ($\rightarrow$) and equivalence ($\equiv$), plus universal ($\forall$) and existential ($\exists$) quantifiers. Some extended quantifiers can be introduced:
     - Exactly one quantifier: $\exists!$, $(\exists!x)\phi(x) \equiv (\exists x)(\phi(x) \wedge \neg(\exists y)(\phi(y) \wedge y \neq; x))$.
       E.g.: $(\forall x{:}Person)(\exists!y{:}Mother)has(x,y)$.
     - Unique existential quantifier: $\exists!!$, $(\forall x)(\exists!!y)\psi(x,y) \equiv (\forall x)(\exists!y)(\psi(x,y) \wedge \neg(\exists z)(\psi(z,y) \wedge z \neq; x))$.
       E.g.: $(\forall x{:}Person)(\exists!!y{:}SocialSecurityNum)has(x,y)$.
   - *Horn-clause logic* is a subset of FOL, it has not disjunction ($\vee$) in implication ($\rightarrow$) conclusions, this is on the right of the implication.
   - *Propositional logic* is also a subset of FOL; it includes Boolean operators but no quantifiers.
3. **Proof theory**: another kind of change is to restrict or to extend permissible proofs.
   - *Linear logic* restricts proofs allowing each propositions to be used only once in a proof.
   - *Non-monotonic logic*, instead, extends proof procedures by introducing default assumptions if they are consistent with what is currently known.
4. **Model theory**: it defines how the logic is related to world, i.e. the denotation or truth-value of logic statements.
   - *FOL* has two values to characterise this relation, true or false.
   - *Tree-valued logic* introduces a third value, unknown, which characterises statements whose denotation cannot be determined.

- *Fuzzy logic* is multi-valued with characterisations, certainty factors, from 0.0, certainty true, to 1.0, certainty false.

5. **Ontology**: An uninterpreted logic has no predefined predicates for representing any subject; its only symbols are quantifiers, Boolean operators and variables. In practice, to provide some building blocs that facilitate their use, some logics include predefined predicates and axioms, their built-in ontologies.

   - *Set theory* is used to provide mathematical foundations.
   - *Temporal* and *dynamic logics* provide time ontologies.

6. **Metalanguage**: this is a language about language. Metalanguages are used for defining, modifying or extending any language.

   - *FOL* can be used as a metalanguage for any version of logic, including itself.
   - *Context-free grammar* is a version of Horn-clause used as a metalanguage for defining the syntax of languages. In general, grammars are equivalent to some subset of FOL used as a metalanguage.
   - *Modal logic* is a FOL metalanguage extension to accommodate modal auxiliary verbs, not used to talk about the way the worlds is, but the way it may, can, must, should, would or could be. Basic modal logics assume two modes:
     p necessarily true (must): $\Box p \equiv \neg\Diamond\neg p$.
     p possibly true (can): $\Diamond p \equiv \neg\Box\neg p$.

System T is a version of modal logic. Each version is defined by the assumed axioms that make them more or less strong. This version assumes, among others, one important axiom:

$\Box p \rightarrow p$ (anything that is necessarily true is true).
For instance, this axiom used in database theory to define constraints.

Deontic logic is a weak version of Modal logic, for instance weaker than System T. It is particularly appropriate for legal domains. Modal operators are interpreted as obligation ($\Box$) and permission ($\Diamond$). Moreover, the previous axiom is not applicable as laws can be violated, i.e. it is not true that anything obligatorily true is true. A weaker version of it must be assumed:

$\Box p \rightarrow \Diamond p$ (anything obligatorily true is permissibly true).

Higher-order logic is a metalanguage extension to FOL plus an ontology for relations. Thus, despite FOL quantifiers that can only range over simple individuals and names of relations and functions, HOL quantifiers can also range over relations and functions, i.e. domains of relations and functions constructed out of domains of individuals.

These can also be seen in the differences in equality theory. In first-order semantics, equality between predicates is intensional, names level, while higher-order equality is extensional, over the predicate domain. Some examples of Higher-order logics are:

Second-order logic is supplemented with an ontology for all possible relations among simple individuals.

Third-order logic includes an ontology for all relations of relations and etcetera for logics of order above.

Compared to FOL, HOL has greater expressive power but, opposite to FOL, it is incomplete [53] and thus lacks practical computable models. More about this issue is presented in the Inference section 2.8.

However, some syntactical tricks can be used in the FOL domain to enjoy some higher-order logic features. Instead of using higher-order semantics that allow variables range over relations, first-order semantics can be combined with higher-order syntax. Higher-order syntax allows variables to appear in places where normally predicate or function symbols do and, therefore, assertions about assertions are possible. Altogether, in latter instance, this is only what is informally called "syntactic sugar". Some kind of stratified reasoning is needed to finally avoid HOL complexity. FOL-like reasoning is carried out separately for the object level and the meta-level [74] provided by higher-order syntax.

Reification is a formal trick of this type. For each predicate, a constant that stands for it is created. These constants are used in axioms. However, since they are constants, the resulting theory is first-order. This is useful for expressing control knowledge; facts are reified so one can say when they are relevant. Reification can also be used to reify modal logic semantics and drop reasoning to first-order.

Frame logic [80] is an example of this kind of logics; it combines first-order semantics with higher-order syntax. In contrast, FOL has both first-order syntax and semantics.

2.7 Semantics

To better explain the term semantics, it is going to be situated in the context of semiotics, the general theory of representations. Moreover, as altogether is used by agents, the view is going to be completed with its relation to agents from the systems point of view. This would complement the view of knowledge from systems theory presented in the Tacit Knowledge section 2.1.

2.7.1 Semiotics

Semiotics studies signs, that comprise icons, tokens, symbols, etc., and thus the complete representations range. It covers them in general: their use in language and reasoning and their relationships to the world, to the agents who use them, and to each other. Therefore, it comprises all languages, informal and formal, presented in previous sections.

Human languages are a specific case of semiotics. A language is a system of conventional spoken or written symbols by means of which human beings, as members of a social group and participants in its culture, communicate.

Formal languages, like logics, are also studied by semiotics. Logics are also based on symbols. The difference is that some restrictions are imposed on syntax and semantics. These restrictions reduce their expressiveness but also ambiguity and thus facilitate their automation.

Semiotics has three dimensions that cover specific aspects of signs:

1. Syntax: it deals with relations among tokens and the production of new ones.
2. Semantics: it studies how agents interpret tokens and relate them to the things they stand for.
3. Pragmatics: it analyses the repercussions of token interpretations for the agent in the environment. It includes a purpose, represented as goals or desires, which ultimate criterion is to aid system in survival.

In Figure 2.4 these three terms are situated in the systems theory view of agents previously presented in the Tacit Knowledge section 2.1.

2.7.2 Sense and reference

As a part of semiotics introduced in previous section, semantics deals with how knowledge representations are related by agents to the things they stand for. This description must be further detailed. At a first glance, it only captures the part of semantics that relates an agent with its environment. This facet should be complemented, as will be demonstrated soon, with a supplementary one that will complement the full range of processes where semantics are involved. These more detailed aspects of semantics are denotational and representational semantics [18]:

- **Denotational - Reference**: the denotational aspect is outward looking. It is the study of the relation of symbols to what they stand for in the world. Reference corresponds to the

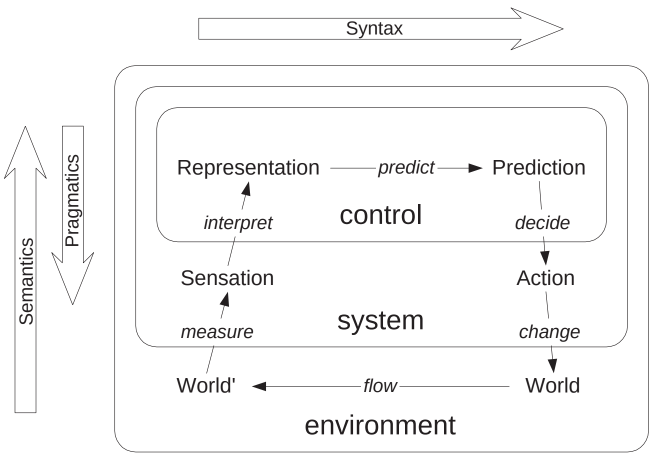

Figure 2.4: The semiotics of the systems view of knowledge

denotation and it is the extension of an expression, what it stands for on a given situation of its use.

- **Representational - Sense**: on the other hand, the representational aspect is inward looking. It considers how contents are mentally represented and corresponds to the sense or intension of an expression. The sense is what enables us to communicate with each other. It is inter-subjective, and thus objective inside a community of users. Therefore, it is not the individual subjective mental representation. It is the information content we grasp in understanding a sentence, the meaning.

This two-sided view of semantics gives birth to the sense and reference distinction shown in the Meaning Triangle Figure 2.5.

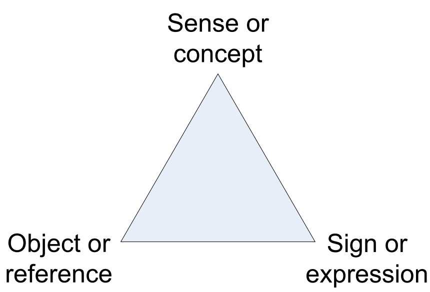

Figure 2.5: The Meaning Triangle

Considering the previous distinction, an expression stands for its reference, not in all respects, but in relation to a sort of idea, its sense [92]. The sense-reference distinction allows dealing with semantics extreme cases, like multiple expressions with the same reference or expressions with non-existent reference, e.g. fiction entities.

An example of the first case is shown in Table 2.1, where other examples about more elaborated expressions are also shown.

Expression	**Reference**	**Sense**
Nominal Phrases *"The morning star"*	Individuals *Venus*	Individual concepts *The concept of the star that disappears last in the morning*
Verbal Phrases *"Is Italian"*	Classes of individuals *The Italians*	Concepts *The concept of being Italian*
Sentences *"Pavarotti is Italian"*	True or false *True*	Thoughts *The though that Pavarotti is Italian.*

Table 2.1: Meaning Triangle examples

The expression "the morning star" is used about the last star visible in the morning; this is the sense of the expression. However, the reference is the same as that for the expression "the evening start", the Venus planet. On the other hand, the meaning for the later is the star that appears first in the evening.

Sentences are more complex expressions that denote a proposition or world state of affairs. Denotational semantics makes possible to determine the validity of this proposition, i.e. if the state of affairs holds or not in the current state of the world where interpretation takes place.

2.7.3 Automatic semantics

It must be remarked that common knowledge representation systems work at the representational level. They manage pieces of information and relate them to senses previously established by knowledge representation means. These sense definitions are mainly captured by ontologies, one of the components of knowledge representation. The other non-computational component, logic, may also capture some representational semantics as built-in ontologies.

Therefore, representational semantics operate only at the abstract level. A great part of the meaning can be captured at this level. This can be done defining concepts, conceptual relations among concepts and combining them to capture expressions that are more complex. However, to acquire its full semantics, a knowledge representation must be grounded, i.e. connected to the world.

At the denotational level, knowledge representation systems are only able to maintain truth if they use valid reasoning methods and start from valid premises, more details in next section. The denotational semantics of knowledge representation languages are also called language semantics. They describe how to compute expression interpretations from previous interpretations of their constituents. Therefore, some preliminary grounding at the starting point is needed. Moreover, if truth is not preserved by reasoning methods, further interpretation is necessary to check conclusions validity.

In order to automate the denotational level, machines must have some kind of external world sensors that allow them to relate expressions and concepts to external objects. These issues are studied applying robotics and artificial intelligence techniques and lay outside the scope of the research work captured in this memory.

2.8 Inference

Inference is what makes a logic more than a notation. Inference rules determine how one pattern can be generated from another. Thus, new pieces of knowledge can be added based on previous ones [99]. The final objective is to capture how agents in general reason about what they know. There exist different kinds of reasoning and this leads to different inference types: deduction, abduction and induction.

2.8.1 Deduction

It is also known as logical inference because deduction is the type of reasoning that logics try to capture. The more important characteristic of deduction is that it preserves truth as determined by semantics. From true premises, it guarantees a true conclusion.

Logics that support this kind of reasoning are called sound logics. More details about this are presented in Table 2.2.

These are the rules of inference for propositional logic, i.e. without quantifiers:

1. *Modus ponens*: from p and p→q, derive q.
2. *Modus tollens*: from ¬q and p→q, derive ¬p.
3. *Hypothetical syllogism*: from p→q and q→r, derive p→r.

Semantic tests, stated in terms of the entailment operator $\models$, provide criteria for evaluating the rules of inference. Entailment operates at the denotational level, while inference operates at the referential level. Rules of inference define the provability operator $\vdash$, i.e. that something is provable. Semantic entailment is more fundamental than provability because it derives the truth of formulas from facts about the world. Provability depends on the rules of inference of a particular version of logic, and those rules must be justified in terms of entailment. Two desirable properties of inference are: **Soundness** means that everything provable is true. Rules of inference are sound if provability $\vdash$ preserves truth as determined by semantic entailment $\models$. $(\forall$s:Situation$)(\forall$p,q:Proposition$)($s$\models$ p $\rightarrow$ (p$\vdash$ q $\rightarrow$ s$\models$ q)). **Completeness** is the converse of soundness. Everything true is provable. $(\forall$s:Situation$)(\forall$p,q:Proposition$)(($s$\models$ p $\rightarrow$ s$\models$ q) $\rightarrow$ p$\vdash$ q)). For FOL, the distinction between $\vdash$ and $\models$ can be ignored because soundness and completeness guarantee that they are equivalent. For other versions of logic, however, they must be carefully distinguished. Kurt Gödel proved in 1931 [53] the incompleteness of higher-order logic by finding propositions entailed by $\models$ that are not provable by $\vdash$. Non-monotonic logic is not even sound because it makes default assumptions. Instead of preserving truth, the non-monotonic rules of inference preserve only the weaker property of consistency: all true statements must be consistent, but not all consistent statements are true.

Table 2.2: Soundness and Completeness

4. *Disjunctive syllogism*: from $p \vee q$ and $\neg p$, derive q.
5. *Conjunction*: from p and q, derive $p \wedge q$.
6. *Addition*: from p, derive $p \vee q$.
7. *Subtraction*: from $p \wedge q$, derive p.

These ones are the inference rules for deduction with quantifiers. Together with the rules for propositional logic conform FOL rules of deduction:

1. *Universal instantiation*: from $(\forall x)\phi(x)$, derive $\phi(c)$, where c is any constant.
2. *Existential instantiation*: from $\phi(c)$, derive $(\exists x)\phi(x)$.
3. *Dropping quantifiers*: if x is not free in ϕ (i.e. it is bound), then from $(\exists x)\phi$ derive ϕ, and from $(\forall x)\phi$ derive ϕ.
4. *Adding quantifiers*: from ϕ derive $(\forall x)\phi$ or $(\exists x)\phi$, where x is any variable.
5. *Substituting equals for equals*: from terms s and t where s = t, derive $\phi(t)$ from $\phi(s)$.

Full deduction, though complete, is computationally an expensive process. Therefore, when it is implemented, some tricks are applied to make this process practical. Two general approaches are forward-chaining and backward-chaining. They consider only subsets of logic deduction.

- **Forward-chaining**: it is centred on the *Modus ponens* rule of inference, $p \wedge p \rightarrow q \vdash q$. Forward-chaining is applied at assertion time, i.e. when new knowledge is introduced into a knowledge base. A knowledge base is the part of a knowledge system where knowledge is stored. Therefore, it is associated to insertions and modifications.
  One example of forward-chaining implementation is SQL triggers. Another one is inheritance that propagates characteristics from broader to narrower concepts.

- **Backward-chaining**: it considers the *Modus tollens* rule of inference, $\neg q \wedge p \rightarrow q \vdash \neg p$. Backward-chaining is used at query time, i.e. when answering questions posed to a knowledge system, which answers them from what there is in the knowledge base.
  For instance, backward-chaining is used to solve SQL queries and in the Prolog programming language. Another example is resolution. It is a complete and sound form of backward-chaining deduction. Usually, forward-chaining implementations apply so many restrictions that they sacrifice completeness in order to improve efficiency.

2.8.2 Abduction

This inference type is not a legal inference because it allows false conclusions. Abduction is an explanation generation process. Some chunks of knowledge are selected, evaluated against the problem at hand and finally packaged into a theory.

Abduction may be performed at various levels of complexity [106]:

1. Reuse: associatively search a predefined theory and reuse it for the current problem.
2. Revise: find a theory that approximately matches the problem at hand and apply revision techniques to tailor it for the current situation.
3. Combine: search fragments of knowledge and repeatedly perform revision steps to combine them into a complete theory.

It can be summarised as: from $b \wedge a \rightarrow b$ perhaps a, i.e. it seeks an explanation for b being true.

2.8.3 Induction

It is the inference process involved in learning, which tries to anticipate how things will act. From series of facts, a generalisation is concluded. Induction is not a valid inference because it does not guarantee truth. Thus, it needs retraction upon contradiction, i.e. when an instance that contradicts the generalisation is found.

It can be expressed as: from $P(a), P(b), \ldots$ conclude $(\forall x)P(x)$.

2.8.4 Analogy

Another kind of inference can be considered, analogy. It is a combination of second-order induction plus deduction. However, it does not preserve truth, and, indeed, not even falsify. Yet, analogy is very useful for argumentation, scientific discovery, case-based reasoning or planning.

Analogy can be summarised as: from $P(a) \rightarrow P(b) \wedge R(a) \rightarrow R(b)$ perhaps $Q(a) \rightarrow Q(b)$.

2.9 Ontology

Ontology has been a field of philosophy since Aristotle and from its beginnings, it has been characterised as a study of existence, a compendium of all there is in the world.

Nowadays, it has evolved in great measure in the computer science and artificial intelligence fields. Currently, ontologies are viewed as a shared and common understanding of a domain that can be communicated between people and heterogeneous and distributed application systems. A detailed description is presented Figure 2.6.

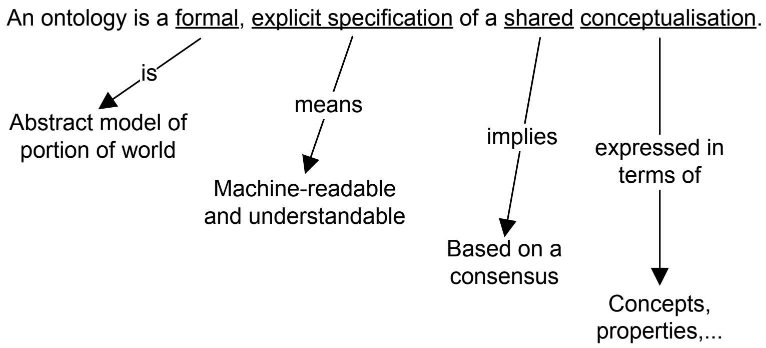

Figure 2.6: Ontology definition [96]

Firstly, they were extensively used in artificial intelligence to facilitate knowledge sharing and reuse. Currently, their use is expanding to other disciplines related to information technologies. In the future, they may play a major role in supporting the information exchange processes, as they provide a shared and common understanding of a domain.

Ontologies are constructed using knowledge representation languages and logics. This allows that automatic devices make informed domain-dependent reasoning using the knowledge captured by ontologies.

As was commented in Automatic Semantics section 2.7.3, a great part of expressions meaning can be captured combining simpler concepts and conceptual relations, i.e. ontologies. At the end, some preliminary set of simple concepts and relations is found. This set must have a rich semantic grounding in order to make powerful and valid automatic reasoning. Moreover, if it is shared among a great community, it may permit a great level of understanding.

The kind of ontology presented in the previous paragraph is called top level or upper ontology. More details about upper ontologies are given in the Upper Ontologies section 2.9.1.

On the other hand, there are domain-level ontologies. They are based on upper ontologies but deal with more specific domains. Some example domains and the corresponding domain ontologies are:

- Genomics and proteomics: Gene Ontology™Consortium, http://www.geneontology.org

- Enterprise: The Enterprise Ontology,
  http://www.aiai.ed.ac.uk/~entprise/enterprise/ontology.html
- Medicine: Unified Medical Language System,
  http://www.nlm.nih.gov/research/umls
- Research: Semantic Web Research Community Ontology,
  http://ontobroker.semanticweb.org/ontos/swrc.html
- Education: Educational Modelling Language,
  http://eml.ou.nl
- Business: Process Specification Language,
  http://ats.nist.gov/psl

2.9.1 Upper Ontologies

Upper ontologies, also known as foundational or top-level ontologies, try to formalise the more general concepts in our conception of the world. There are many attempts to produce upper ontologies.

SUMO

Many upper ontology initiatives have joined in the IEEE SUO effort (Standard Upper Ontology[1]). One result of this working group is the SUMO ontology (Suggested Upper Merged Ontology[2]) [84]. Some of the general topics covered in the SUMO include:

- Structural concepts such as instance and subclass
- General types of objects and processes
- Abstractions including set theory, attributes, and relations
- Numbers and measures
- Temporal concepts, such as duration

[1] http://suo.ieee.org
[2] http://www.ontologyportal.org

- Parts and wholes
- Basic semiotic relations
- Agency and intentionality

DOLCE

The Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) [41] is another upper ontology that has a clear cognitive bias, in the sense that it aims at capturing the ontological categories underlying natural language and human commonsense.

Endurants and Perdurants DOLCE is based on a fundamental distinction between enduring and perduring entities, i.e. continuants and occurrents, a distinction motivated by our cognitive bias. Classically, the difference between enduring and perduring entities is related to their behaviour in time. Endurants are wholly present, i.e. all their proper parts are present, at any time they are present. Perdurants, on the other hand, just extend in time by accumulating different temporal parts, so that, at any time they are present, they are only partially present, in the sense that some of their proper temporal parts, e.g. their previous or future phases, may be not present. E.g., the piece of paper you are reading now is wholly present, while some temporal parts of your reading are not present any more.

In DOLCE, the main relation between endurants and perdurants is that of participation: an endurant "lives" in time by participating in some perdurants. For example, a person, which is an endurant, may participate in a discussion, which is a perdurant. A person's life is also a perdurant, in which a person participates throughout its all duration.

LRI-Core

LRI-Core [16] is a core ontology that covers the main concepts that are common to all legal domains. These domains have a predominant common-sense character and typical legal concepts such as norm, role, responsibility, contract, etc. have a grounding in abstract common-sense conceptualizations. This common sense grounding is lacking in various upper or foundational ontologies developed thus far.

One of the most important design principles that follows from the common-sense stance in developing the LRI-Core is cognitive plausibility. From this perspective, knowledge about the physical world, with the central notions of object and process is taken as a basis for a mental and abstract worlds. The intentional stance that differentiates the physical world from the mental world is also the basis for the creation of a behavioural world of roles. In summary, LRI-Core starts with four main categories: physical concepts, mental concepts, roles, and abstract concepts. A fifth category consists of terms for occurrences, which are used to talk about instances (situations) in a generic way.

2.10 KR Technologies

Knowledge representation comprises logic, ontology and computation. Previous sections have centred on the first two components. At most, computation has been considered in relation to complexity issues. This section considers knowledge representation as a conjunction of these three components. Different implementations, i.e. conjunctions of logic, ontology and computational resources, are presented. These implementations are the last step towards putting knowledge representation into practice.

2.10.1 Logic programming

The most relevant feature of logic programming technologies is that they implement restricted versions of logic to improve performance. Generally, they use backwards-chaining implementation of inference.

Logic programming has evolved from research on theorem provers. Theorem provers use full first-order logic and implement Robinson's resolution rule, a complete and sound form of backward-chaining deduction. They are used mainly for mathematical and scientific problems.

On the contrary, logic-programming languages are used when greater performance is needed. They usually sacrifice completeness to reach this goal. However, both approaches are based on backward chaining a thus best suited to question answering.

Prolog

It is a widely used logic programming language. It imposes some restrictions to its logic component. Horn-clause logic is used so it has not disjunction ($\vee$) in implication ($\rightarrow$) conclusions. Moreover, there is no negation in Prolog. However, it is supported at the metalevel by negation as failure. When something cannot be demonstrated with what is known, it is considered false:

Negation as failure: if fail to find answer assume falsehood, $(\forall x)P(x) \equiv \neg(\exists x)\neg P(x)$.

The inference mechanism is not complete. It implements backward chaining with depth first search. When the search is trapped in a cul-de-sac, backtracking is applied, i.e. last steps are undone until a new search path.

Finally, Prolog incorporates some built-in predicates and functions that provide useful primitives and non-logical programming facilities, e.g. computer input/output management. They are the facilities and building blocks over which logic programs and personalised predicates and functions are defined. They conform the ontology that captures the knowledge structures the Prolog knowledge base over which logic programs work.

2.10.2 Production systems

Like programming languages, production systems use implication as the primary representation element. They operate with a forward chaining control structure that operates iteratively. When the premises of an implication, known as rule, are satisfied by facts in the knowledge base, it is fired. Therefore, they are particularly suited to model reactive behaviours. An example of production system rule is presented in Table 2.3.

not carrying ∧ nearby(x) ∧ grain(x) → pickup(x) ∧ carrying(x) ∧ moveRandom
carrying(x) ∧ nearby(y) ∧ grain(y) → drop(x) ∧ not carrying(x) ∧ moveRandom

Table 2.3: Production System rules example

Firing a rule results in interpreting its consequents. They are interpreted as actions and not as mere logical conclusions. These actions, among others, comprise knowledge base insertions and deletions. Some production systems have mechanisms to resolve cases where many rules can be fired simultaneously. For instance, the may resolve them implementing rule precedence or a non-deterministic selection behaviour.

Jess

Jess is production system implemented with Java. It is inspired in a previous production system called CLIPS. Jess implements a rete algorithm to improve rule-matching performance, the most important aspect of this kind of knowledge representation systems. In short, the algorithm maintains pointers to partial rule matches from data elements. This directly relates rules to data whose changes might affect them. Therefore, when some data is changed, it is quite direct to know which rules might then get all their firing conditions satisfied.

2.10.3 Semantic Networks

They are particularly suited to model static world knowledge. World objects and classes of objects are modelled as graph nodes and binary relations among them are captured as edges between nodes. There are different types of edges. Remarkably, a special type of edges defines taxonomical relations between nodes, i.e. subsumption of classes and object-class membership.

The taxonomy supports a built-in fast inference method, inheritance, to reason about generic and specific object features. The general graph structure can be used to efficiently reason about inheritance and to locate information.

Their greatest problem has been that they have lacked consensus semantics for a long time. Currently, they have been completely formalised as a subset of FOL.

HTML

The HTML language of World Wide Web can be viewed as constructing a global Semantic Network, with pages, links between pages and very limited set of link types. The only distinction is between the external link . . . and the link to embedded images <img src=”. . . ”>.

Frames

Frames are an evolution of semantic networks. They add procedural attachments to the node and edge structure of semantic networks. Altogether, the resulting framework and modelling paradigm has evolved into the object oriented programming paradigm.

This new paradigm has had great acceptance, for instance Sun’s Java object oriented programming language. Objects and classes are nodes and relations are modelled as object references stored in object and class variables. The taxonomical relations are built-in in Java language. Subsumption is declared as “subclass extends class” constructs in class definitions. Object-class membership is stated when a new object is created with the “object = new Class” construct. The procedural attachments are represented as class methods. Their behaviour is defined with the procedural part of Java in class definitions.

2.10.4 Description Logics

They are a formalisation of Semantic Networks that allows them to be seen as sub-languages of predicate logic. They are considered an important formalism unifying and giving a logical basis to the well-known traditions of frame-based systems, semantic networks and object-oriented representations.

Special emphasis is placed in concept definitions, roles and taxonomy building constructs. Concepts describe the common properties of a collection of individuals and roles are interpreted as binary relations between objects.

Description Logics allow specifying a terminological hierarchy using a restricted set of first order formulas. Restrictions make that Description Logics usually have nice computational properties. They are often decidable and tractable, or at least seem to have nice average computational properties, but the inference services are restricted to subsumption and classification.

Subsumption means, given formulae describing classes, the classifier associated with certain description logic will place them inside a hierarchy. On the other hand, classification means that given an instance description, the classifier will determine the most specific classes to which the particular instance belongs.

Each description logic defines also a number of language constructs, such as intersection, union, role quantification, etc., that can be used to define new concepts and roles.

FaCT

Fast Classification of Terminologies (FaCT) is a Description Logic classifier. The FaCT system includes two reasoners, one for the description logic SHF (ALC augmented with transitive roles, functional roles and a role hierarchy) and the other for the logic SHIQ (SHF augmented with inverse roles and qualified number restrictions). Both of them are implement using sound and complete tableaux algorithms. This kind of algorithms is specially suited for subsumption computation and has become the standard for Description Logic systems [60]. More details about different description logics are presented in Figure 2.7.

2.10.5 Conceptual Graphs

Conceptual Graphs were developed from Existential Graphs [91] and Semantic Networks. They are under standardisation process [4] in conjunction with KIF (Knowledge Interchange Format) [47]. Both are different ways of representing FOL. KIF is an attempt to standardise a linear syntax of FOL while Conceptual Graphs provide a diagrammatic syntax. A comparative of them is presented in Figure 2.8 and Table 2.4.

Although Conceptual Graphs are equivalent to FOL, their graph orientation provides many

Construct	Syntax	Language
concept	A	FL^-
role name	R	
conjunction	$C \cap D$	
value restriction	$\forall R.C$	
existensial quantification	$\exists R$	
top	T	AL^*
bottom	$\perp$	
negation (C)	$\neg A \; \neg C$	
disjunction (U)	$C \cup D$	
existential restriction (E)	$\exists R.C$	
number restrictions (N)	$(\geq n R) \; (\leq n R)$	
collection of individuals (O)	$\{a_1 \dots a_n\}$	
role heirarchy	$R \subseteq S$	H
inverse role	R^-	I
qualified number restriction	$(\geq n R.C) \; (\leq n R.C)$	Q

with transitive roles also called S and hence SHIN, SHIQ

Figure 2.7: Description Logics languages and their characteristics, from [97]

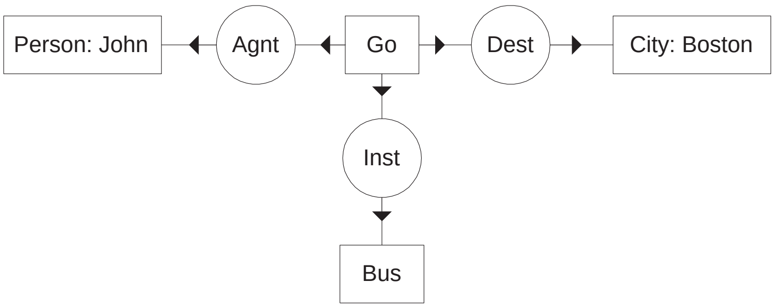

Figure 2.8: Conceptual Graph for "John goes to Boston by bus"

exists ((?x Go)(?y Person)(?z City)(?w Bus))
(and (Name ?y John) (Name ?z Boston)
(Agnt ?x ?y) (Dest ?x ?z) (Inst ?x ?w)))

$(\exists x{:}Go)(\exists y{:}Person)(\exists z{:}City)(\exists w{:}Bus)$
$(name(y,'John') \wedge name(z,'Boston') \wedge$
$agnt(x,y) \wedge dest(x,z) \wedge inst(x,w))$

Table 2.4: KIF and FOL equivalent representations for the example conceptual graph in Figure 2.8

features that make them especially useful. They express meaning in a form that is logically precise, humanly readable, and computationally tractable. With a direct mapping to language, conceptual graphs serve as an intermediate language for translating computer-oriented formalisms to and from natural languages. With their graphic representation, they serve as a readable, but formal design and specification language.

Moreover, there have new inference facilities. Existential Graphs provide a more simple set of rules to perform deduction. Other kinds of inference are also benefited. Graph structure can be used to implement efficient algorithms to reason with analogy [77].

2.11 Knowledge into Practice

When finally Knowledge Representation is put into practice by applying knowledge representation technologies, two disciplines can be considered: Knowledge Engineering and Knowledge Management.

2.11.1 Knowledge Engineering

It is a terms that refers to Knowledge Representation used for some purpose. Its relation to knowledge representation can be compared to the relation of Mathematics or Physics to Engineering.

2.11.2 Knowledge Management

It is a broad discipline that usually involves Knowledge Representation techniques. It ranges people, organisational processes, business strategies and Information Technologies. Knowledge Management could be defined as the development of best practices to add value to a company through maximum use of the data, information and knowledge within. It should also harness potential knowledge and tacit knowledge, with the intention of converting them into explicit knowledge, to be then made accessible as shareable and re-usable company assets.

Information Systems process information without engaging the users. Knowledge Management Systems help users understand information and, contrary to Information Systems, they include users perspective, i.e. the relation between the piece of information and users knowledge. This includes the situation where the user develops the information need and the analysis of the usage of the same information once it has been obtained and interpreted by the user [2].

Chapter 3

Web Technologies

The intention of this section is not to make a complete state of the art of web technologies. The purpose of this memory is to recompile the research work related to Semantic Web and its perspectives. Therefore, the focus of this section is on presenting current web technologies and the actual web scenario. They will be used as the building blocks and the starting point from where to explore the development of new solutions. The premise is that the Web can be completed with the contributions of a new proposal that may solve some of its problems, the Semantic Web.

3.1 Introduction

The World Wide Web appeared as the result of the need to integrated many disparate information systems. The approach was to form an abstract space in which the differences between them did not exist. The Web had to include all information of any sort on any system.

Back in 1989, when the World Wide Web was first conceived, many different information systems existed. They ran on different sorts of computers, each running different operating systems, connected by different networks, and using quite different programs to give to the user very different ways of accessing information. Thus, while the information on two systems might be very relevant, the path between them was very long.

In fact, the Internet already existed. Each of the computer systems was very likely to be connected to some sort of network, which very likely was in turn connected to another network. There was a path from a bit of data on one computer through a series of networks to the other computer. Thus, there was no reason why software and hardware barriers to communication should exist.

Software and device independence was achieved in both sides, the client accessing information and the server offering it.

The Web started to be deployed with only a common idea needed to tie it all together, the URI (Universal Resource Identifier) that identified a document in the WWW abstract space. From that cascaded a series of protocols, such as HTTP (HyperText Transport Protocol), and data formats, such as HTML (HyperText Markup Language), which allowed computers to exchange information, mapping their own local formats into standards that provided global interoperability.

The abstract space of the World Wide Web is based on the distributed hypermedia paradigm that is the synthesis of three ideas [57]:

1. **Hypertext**: it refers to the fact that Web documents are cross-referenced by hotlinks. They are highlighted sections or phrases in the text, which can be selected by the user, calling up an associated document with more information about that highlighted.
2. **Multimedia**: this means that documents can present their information in any modality or format available. For instance, formatted text, drawings, sound, photos, movies, 3-D virtual reality scenes, or any combination of these.
3. **Distribution**: the WWW is distributed because linked documents can reside on different computers, maintained by different people, in different parts of the world. With good network connections, the time needed to transfer a document from another continent is not noticeably different from the time it takes to transfer a document from the neighbouring office. This makes it possible to transparently integrate information on a global scale.

The World Wide Web is now established as a unified interface to the Internet computer network. This universal acceptance is due to the Web is extremely simple, but a powerful way of representing networked information. Its simplicity comes from the premise of providing the maximum freedom to its users. It is resumed in the principle anyone can say anything about anything in the WWW.

For instance, this can lead to links that point to non-existent documents. This kind of situations was initially seen as inappropriate in an information system. However, this lack of strong constraints has turned to be the Web's most important reason for its success.

3.2 Building blocks

From the previous introduction, a set of Web building blocks can be extracted. They conform a quite simple basis over which the whole World Wide Web has evolved.

3.2.1 URI

They play in the Web more or less the same role than IP (Internet Protocol) addresses in the Internet. An IP identifies a device in the global framework of the Internet; an URI is a global document identifier [83].

Due to the original uses of URI in the starting Web, URI as best known for the subset called URL (Universal Resource Locator). This subset is intended for resource location, i.e. how to reach a document. On the other hand, the other subset of URI, URN (Universal Resource Name), is exclusively used for resource identification. Thus, URI is the generic term for both, resource location and/or identification in the World Wide Web.

For examples URIs include,

- URNs, like *urn:isbn:84-85081-95-1* that identifies a book reusing ISBN (International Standard Book Number), and
- URLs, like *http://www.server.org/video.mpg,* the location from where a video might be retrieved as Web links are not required to be valid, i.e. point to an existent resource.

3.2.2 HTTP

It is the WWW communications protocol. HTTP defines how to carry on conversations between Web clients and servers in order to move data across the Web. HTTP is a simple stateless protocol reduced to the interchange of request and response messages between the client and the server [36]:

- **Requests**: they are sent from client to server. The message main parts are an URI and a command. The URI identifies the resource to which the client requests the command to be applied. The most important commands are GET to retrieve the URI-identified entity from the server, and POST/PUT commands to request the server to accept the entity enclosed in the message under the provided URI in the former or to store it as the provided URI in the latter.
- **Responses**: after receiving and interpreting a request message, a server responds with an HTTP response message. It includes information about the outcome of the request and, if some entity was requested, it is included in the body of the message.

3.2.3 HTML

World Wide Web documents have this data format. This format defines how to pack into web documents textual information, external multimedia resources and interactive links to other web documents. Therefore, HTML provides the first two features of the Web presented in the previous section, hypertext and multimedia.

Like the Web, its data format has also suffered rapid evolution that has eventually carried it to confusing situations where a unified standard was not clear. Recently, a normalisation effort has been completed and the XHTML standard [115] has been produced. It updates HTML using more recent web technologies that connect it to the initiatives presented in the Recent Developments section 3.4.

3.3 Current situation

As was presented in the introduction, the World Wide Web has had an enormous success. The result is that the Web has grown exponentially and it currently has acquired enormous proportions.

This is a good new for its users; there is an immense amount of information and opportunities in the WWW to exploit. However, simple accumulation is not the response. In order to efficiently exploit it and extract its full potential more elaborated mechanisms should be layered over the basic building blocks that the Web provides.

Before entering in an overview of some of the currently proposed mechanisms, some examples of the problems the World Wide Web is encountering are presented. They are introduced through a series of scenarios that illustrate real cases of these problems. These scenarios have been selected because they are relevant, they seem to be explanatory and they have been found and even faced during the research work summarised in this work.

3.3.1 Device proliferation scenario

The great success has been combined with an increasing need of mobility and ubiquity. That is the reason why lastly highly portable devices with web access capabilities have appeared.

This kind of devices has special constraints that do not allow transparent access to the Web as it is now. For instance, size limitations reduce device computation capabilities, screen sizes and available peripherals.

This puts back to focus a previously observed problem that seemed to have been solved. There is not clear separation between content and presentation in the web data format. This was seen during the "browsers" war, more and more features were introduced in the main web clients that required proprietary add-ons to HTML. Very specific features bound the transmitted content and the actual presentation they acquired in the intended browser.

The browsers' problem was more or less resolved, though still now web documents must include specialised commands to tailor their content to the accessing browser. There was possible some consensus but principally it was resolved by "brute force", one of the contestants practically became a de-facto standard.

The current situation is more complicated. The range of clients is greater and differences between them cannot be resolved in practice with ad-hoc methods. Some devices even use HTML inspired data formats that are not compatible, like WML (Wireless Mark-up Language).

Now, a clear effort to separate content and presentation has been initiated and, indeed, many results are yet available. Some technologies that at least try to solve this problem are commented in the Recent Developments section 3.4.

3.3.2 Search engines scenario

The amount of information in the World Wide Web is enormous and it makes complicated that users find what they are looking for. A great effort is necessary to locate relevant results among the many times not very accurate outputs of web search engines.

Search engines rely on mainly syntactic means for content matching with user queries. Matching is based on direct comparison of query keywords and the worlds that appear in web documents.

Due to natural language words ambiguity, due to properties like synonymy or polysemy, syntactic methods are not very accurate. Full processing of natural language is currently not available so, in order to automate high quality content location, other approaches should be considered.

The best option is to carry on some kind of pre-processing of web documents. This pre-processing can be performed with machine support, but human intervention is currently mandatory if reliable results are required. Pre-processing produces metadata. It is data about data, in this case some relevant keywords about a web document. They are associated to web documents so a search engine can retrieve them in order to focus matching, in theory they are selected as the world that best represent documents content.

However, this approach does not solve the ambiguity problem. Both, the keywords from the queries and those from web documents metadata can be ambiguous. A recent attempt to solve that uses mechanisms that allow annotators and requesters to relate the keywords they use. These relations state equivalence, broader or narrower terms, etc. between keywords. Following them, many ambiguities can be overcome. This approach is part of the Semantic Web initiative and is detailed in the Semantic Web section 4.

3.3.3 Business scenario

This is an augmented version of the search engines scenario. In this scenario, not only content location is involved. The Web is evolving to become more than a collection of information. This is a particularly sensitive issue in the business world, which is trying to exploit the Web's full potential.

The augmented web is conformed from what are called web services. They make possible business-to-business interactions through the web. However, thought the business world is being the driving force of the Web Services initiative, the whole web and the full range of its uses can be benefited. More details are presented in the Web Services section 3.4.2.

3.3.4 Copyright scenario

This is a subset of the business scenario. It concentrates on trading copyrighted content in the Web. The objective is to have automated content rights negotiation and mechanism that help to guarantee their fair use.

This rather specific scenario has been highlighted because many of the results presented in this research memory belong to this scenario.

3.4 Recent developments

3.4.1 XML

Extensible Mark-up Language [Yergeau04 **??**] is not an alternative to HTML. They complement themselves as they have different objectives; indeed XML appeared to deal with some features that HTML does not cover. XML is a meta-language for creating mark-up languages to describe data. In contrast to HTML, which describes document structure and visual presentation, XML describes data in a human readable format with no indication of how the data is to be displayed. It centres its attention on content.

Therefore, with XML, content and presentation are effectively separated. It is a database-neutral and device-neutral format. Data marked up in XML can be targeted to different devices using eXtensible Style Language [119]. XSL style sheets connect XML content to their presentations. They conform a template that translates selected XML portions to their output format. Usually, HTML is the output format but there is not restriction on what can be produced. For instance, there are style sheets to produce Acrobat®Portable Document Format or WML documents.

Since XML is truly extensible, rather than a fixed set of elements like HTML, use of XML will eventually eliminate the need for browser developers and middleware tools to add special HTML tags, i.e. extensions.

This is possible because XML is a meta-language used to define other domain or industry-specific languages. To construct a specific XML language, also called a vocabulary, a Document Type Definition is defined. A DTD is essentially a context-free grammar. In other words, a DTD provides the rules that define the elements and structure of the new language. An examples of a DTD is presented in Table 3.1 and some XML of the language defined by this DTD is shown in Table 3.2.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT ADDRESSBOOK (PERSON)*>
<!ELEMENT PERSON (LASTNAME, FIRSTNAME, COMPANY, EMAIL)>
<!ELEMENT LASTNAME (#PCDATA)>
<!ELEMENT FIRSTNAME (#PCDATA)>
<!ELEMENT COMPANY (#PCDATA)>
<!ELEMENT EMAIL (#PCDATA)>
```

Table 3.1: DTD for and Address book XML language

Another way of defining XML languages is using XML schemas. XMLSchema [118] is an evolution of DTD that provides more sophisticated constructs that allow defining XML languages with richer structure and less effort.

3.4.2 Web Services

The current web is mainly a collection of information but does not yet provide support in processing this information, i.e., in using the computer as a computational device. Recent efforts around UDDI, WSDL, and SOAP, all detailed next, try to lift the web to a new level of service. Software programs can be accessed and executed via the web based on the idea of web services. A service can provide information, e.g. a weather forecast service, or it may have an effect in the real world, e.g. an online flight booking service. Web services, opposite to web browsing, change things. They

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ADDRESSBOOK SYSTEM "addressbook.dtd">
<ADDRESSBOOK>
<PERSON>
<LASTNAME>Baggins</LASTNAME>
<FIRSTNAME>Frodo</FIRSTNAME>
<COMPANY>The Fellowship of the Ring</COMPANY>
<EMAIL>frodo@baggings.com</EMAIL>
</PERSON>
<PERSON>
<LASTNAME>Gamgee</LASTNAME>
<FIRSTNAME>Samwise</FIRSTNAME>
<COMPANY>The Fellowship of the Ring</COMPANY>
<EMAIL>sam@ringfellowship.org</EMAIL>
</PERSON>
</ADDRESSBOOK>
```

Table 3.2: XML document containing example data for the Address book XML language in Table 3.1

are remote operations with side effects.

The web is organized around URI, HTML, and HTTP as has been shown in the Building blocks section 3.2. Not surprisingly, web services require a similar infrastructure around three analogous technologies: UDDI, WSDL, and SOAP [31].

UDDI

Universal Description Discovery and Integration [112] provides a mechanism for clients to find web services. Using a UDDI interface, clients can dynamically lookup as well as discover services offered by web service providers.

A UDDI registry has two kinds of clients. Firstly, users that want to publish service descriptions and its usage interfaces. Secondly, users who want to obtain services descriptions of a certain kind and bind programmatically to them using SOAP.

UDDI itself is layered over SOAP and assumes that requests and responses are UDDI objects sent around as SOAP messages.

WSDL

Web Services Description Language [116] defines services as collections of network endpoints or ports. In WSDL the abstract definition of endpoints and messages is separated from their con-

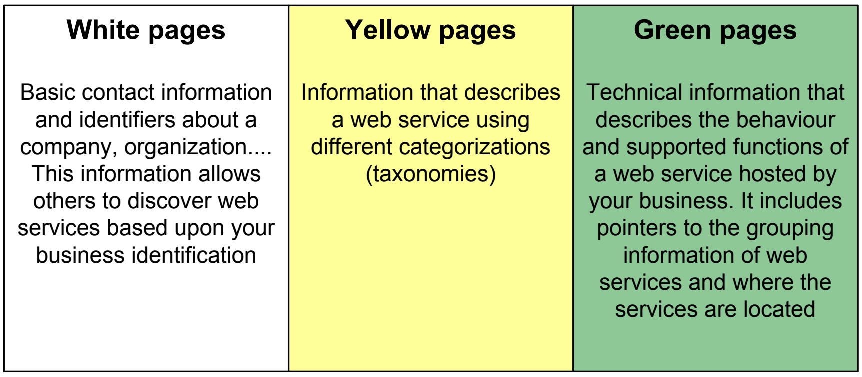

Figure 3.1: UDDI Services

crete network deployment or data format bindings. This allows the reuse of abstract definitions of messages, which are abstract descriptions of the data being exchanged, and port types, which are abstract collections of operations.

The concrete protocol and data format specifications for a particular port type constitute a binding. A port is defined by associating a network address with a binding; a collection of ports defines a service.

SOAP

Simple Object Access Protocol [117] is a message layout specification that defines a uniform way of passing XML-encoded data. It also defines a way to bind to HTTP as the underlying communication protocol, although other communication protocols can be used, for instance electronic mail.

Instead of being document-based, automated B2B interaction requires integration of processes. However, although techniques such as DCOM, RMI and CORBA are successful on the local network, they largely fail when transposed to a web environment. They are rather unwieldy, entail too tight a coupling between components and above all conflict with existing firewall technology.

Replacing this by a simple, lightweight mechanism similar to Remote Procedure Call is the aim of SOAP. Hence SOAP is basically a technology to allow for "Remote Procedure Calls over the web" providing a very simple one-way as well as request and reply mechanism.

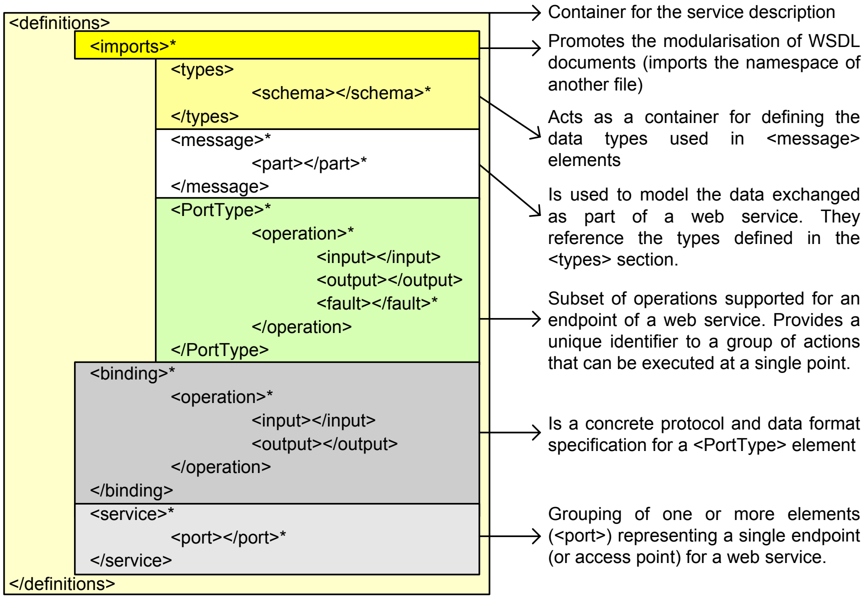

Figure 3.2: Anatomy of a WSDL document

Figure 3.3: Anatomy of a SOAP document

Chapter 4

Semantic Web

The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation

Tim Berners-Lee, James Hendler, Ora Lassila [13]

The Semantic Web is a vision: the idea of having data on the Web defined and linked in a way that it can be used by machines not just for display purposes, but for automation, integration and reuse of data across various applications

Tim Berners-Lee

Semantic Web origins from the premise that the Web is incomplete. It was posed by its inventor, Tim Berners-Lee [10]. Even in his first designs of what the Web should be, there were ideas that did not come into reality in the version of the Web we currently have, which can be called the "Web 1.0".

In 1999, in conjunction with other people interested in creating a new web, Berners-Lee engaged a new trial to get a more complete picture of his initial Web dream. This new attempt was called the Semantic Web and has created a new community of research organised around the Semantic Web Interest Group[1] at the World Wide Web Consortium.

[1] http://www.w3c.org/2001/sw

4.1 Motivation

In the last centuries, we have assisted to an increasing application of technologies to human communication. They range from the press or the telephone, to the digital worldwide publishing environment of the Web [38].

The processing, storage and distribution of information in society has become much more efficient since the introduction of the electronic media. This evolution shows successive stages, which are characterised by the growing complexity of the information processing system.

The first electronic mediums, such as telegraph and telephone, allowed one-to-one communication. Radio and television, the next generation mass media, allowed communication from one to many. The present electronic networks allow many-to-many communication.

On the horizon, next stages seem to be appearing. First, the computer network will become able to learn, that is, change the pattern of its connections. Then, the network will become able to think, that is, autonomously create new information [59].

The Web can be considered the last most revolutionary idea in the human communication domain. It has just appeared, but its rapid evolution has driven it to a situation that seems to require new solutions. A comparison with other communications mediums is shown in Figure 4.1.

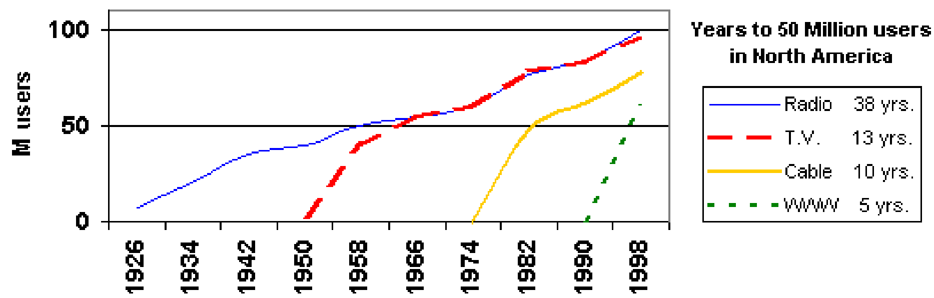

Figure 4.1: Increasing adoption speed of different communication mediums in North America, Morgan Stanley Research

These solutions must resolve the problems that, paradoxically, have emerged due to its huge success. The World Wide Web is far away direct human capabilities, see Figure 4.2, and possible solutions to deal with this enormous amount of information seem to point to the learning and thinking network devised in the previous paragraph.

Currently the Web is merely an information-publishing medium directed towards human consumption. However, as it has evolved and grown, more and more automation solutions have been

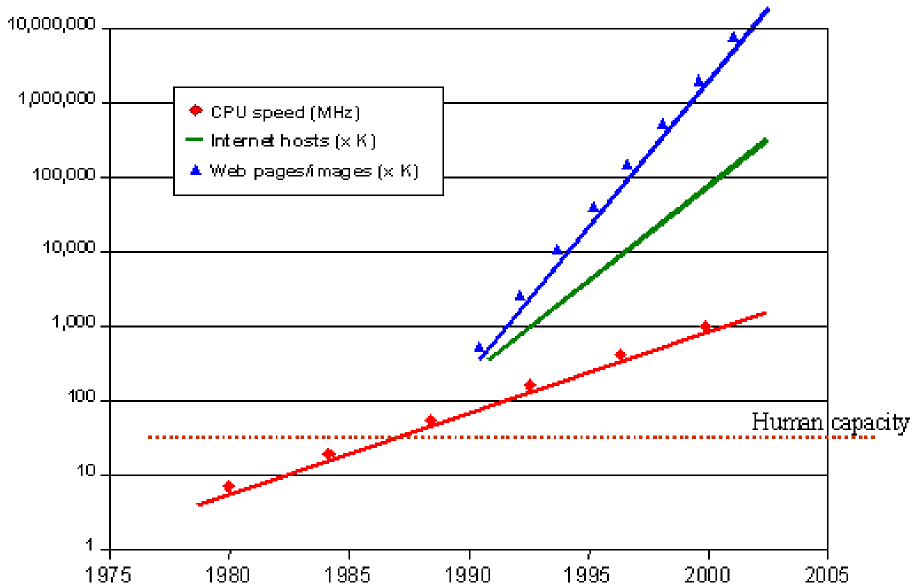

Figure 4.2: The WWW is far away from direct human capabilities, Intel, Commerce Net, AT&T and EC

necessary to allow practical use.

In the Web, computers and networks settle the information space. However, their abilities are not exploited. Until now, communication mediums have not had advanced processing capabilities to manage the information that flow through them.

The Web has this kind of processing capabilities, however, they are not currently exploited. Machines can access only a limited part of the exchanged information, basically its encoding. Thus, advanced information processing can be achieved only following an ad hoc approach.

Only humans are able to catch enough meaning from the information flow to decide how to process it. Programmers have tools to automate these decisions, but they are not expressive enough to provide an automatic framework. Consequently, they are continuously involved in the low-level development issues.

This limitation, common in the Information Technologies domain, is even worst in the Web. It is an open and heterogeneous framework, where millions of interactions happen each day. Some are simple repetitions of previous ones, but most of them are new meetings that require particularised configuration.

On the other hand, limitations of Natural Language technologies do not allow direct access of machines to information. Less ambitious steps are the best solution by now. Computers need explicit help to grasp some of the information meaning.

A first option could be to formalise web content, to use standard forms, sets of words and grammars of universal use in the Web. This is a difficult issue because these formalisations should be agreed globally for each kind of web content.

Another option is to maintain current web contents while providing complementary data about data, i.e. metadata that describes content to machines. However, this does not directly resolve the problem, metadata must be also formalised to be understood by machines.

It would be easier, but not easy, to formalise this at the conceptual level. There are words that refer to the same concept and the same word may take different meanings, refer to different concepts, depending on the context in which it appears. Moreover, in a multilingual Web, conceptual metadata also facilitates multilingual interactions.

Therefore, all this can be accomplished providing semantic annotations of content. These annotations make data meaning explicit by situating it in a conceptual framework. Here, Knowledge Representation techniques are very helpful because there are specialised in formalising information at the conceptual level.

Knowledge Representation is a quite mature discipline, in fact with some millennia of evolution. Lastly, it has largely benefited from technology advances that have allowed its automation. However, it has lacked of big real world application environments where it can be fully applied.

Moreover, when it has been used in smaller domains, rigid operation has been enforced in order to maintain formalism compliance. Thus, due to these two initial restrictions, developed systems evolvability has been seriously compromised and results have been quite predictable.

On the contrary, the Web is a big and heterogeneous environment. On one hand, it provides a huge application space that, on the other hand, forces relaxed Knowledge Representation formalisms in order to obtain practical results.

Rigidity is not appropriate and new ideas based on free evolutionary-like patterns are being considered. The final idea is that, eventually, a world wide distributed knowledge system would emerge, what has also been called the global brain [58], of which we may be seeing its first synapses.

In conclusion, the Semantic Web is new opportunity for both fields, traditional web and Knowledge Representation, where they can operate in conjunction. Benefits from their join application have just started to be explored.

4.2 Tendencies

The question we can pose at this point is why the current Web can be viewed as incomplete. In other worlds, why is the Semantic Web necessary. The reasons that justify this effort started to be depicted in the web scenarios in the Current Situation section 3.3. When the Web's current situation was shown through some conflictive scenarios, what was shown is that there are some tensions. The current evolution of the Web is overcoming its current design. These tension points can be characterized from various perspectives [68]:

1. **Locating Resources**: the way people find things on the Web is evolving from simple free text and keyword search to more sophisticated semantic techniques for both search and navigation. This is now necessary because the Web is so huge that pure syntactical methods do not scale. Moreover, the location problem would be even worst in the future, as the World Wide Web continues growing.

2. **Users**: Web resources are evolving from being primarily intended for human consumption to being intended for use by both humans and machines. There is too much information in the Web for direct human consumption. A higher degree of automation is needed in order to construct meaningful automatic filters between final users and the Web.
   In addition, the great diversity of devices accessing the Web encourages the separation of content and presentation. This facilitates the automation of information flow from primary content to specialised presentations depending on user contexts.

3. **Web Tasks and Services**: the Web is maturing; from its initial uses oriented towards information consumption, it is progressively integrating in society routine. This implies taking profit from web features for other purposes. Therefore, it is evolving from being primarily a place to find things to being a place to do things as well [104].
   One clear example of this is electronic commerce. We can see the enormous possibilities that the Web offers to commercial relationships. However, it seems that the current web is not prepared yet to provide the needed infrastructures to smoothly integrate commerce in this new medium. "Webizing" commerce should not change the way it is carried out, just facilitate it and open new possibilities without closing previous ones.

All of these Web requirements depend in a fundamental way on the idea of semantics. This gives rise to a new integrative perspective along which the Web evolution may be viewed: the Web is evolving from containing information resources that have little or no explicit semantics to having a rich semantic infrastructure. Where, explicit is considered from a machine point of view.

4.3 Built-in Semantics

There is no widespread agreement on exactly what the Semantic Web is, nor exactly what it is for. What poses the problems to define it is the word "semantic", as the word "web" and their connotations are actually almost the common use.

From the introductory cites at the beginning of this section, there is clear emphasis on the information content of the Web being machine usable and associated with more meaning. Here, “machine” refers to computers or computer programs that perform tasks on the Web. These programs are commonly referred to as software agents and are found in Web applications.

The simpler way of making machines aware of the semantics they should manage is the common way software is developed. Web-applications developers hardwire the knowledge into the software so that when the machine runs the software, it does the correct thing with the information.

In this situation, machines already use information on the Web. There are electronic broker agents that make use of the meaning associated with Web content words such as "price", "weight", "destination" and "airport", to name a few.

Armed with a built-in "understanding" of these terms, these so-called shopping agents automatically peruse the Web to find sites with the lowest price for a book or the lowest airfare between two given cities. Therefore, we still lack an adequate characterization of what distinguishes the future Semantic Web from what exists today.

4.4 Explicit Semantics

Despite the current presence in the WWW of built-in semantics, the Semantic Web effort continues to make sense. Indeed, built-in semantics are the problem. The World Wide Web conforms a global communication medium where people and machines meet. If each one has their own built-in semantics, totally isolated from other ones, information flows but communication is impossible.

There are many attempts to make this communication space uniform enough to make possible global understanding. However, this is only viable at reduced scale. Therefore, many standardisation efforts have been done, are being done or are planned. Each one focuses on specific domains where consensus is possible.

People has the ability to grasp others terms semantics. They have a plethora of mechanisms to make themselves understandable and to understand what others say. The most important one is a huge amount of world knowledge. They can connect views and uses of words, or lets say standards. However, this is not easy and it is even worse when the scenario is the Web, there is an immense number of interlocutors from extremely different origins.

Here machines can help to manage these mappings but, first, we must help them understand what they are dealing with. The first step is to make semantics explicit using metadata. Semantic metadata is data about data that is machine-processable. Then, explicit semantics provide the anchor points that make possible semantic interconnections. To effectively build explicit and global semantics the different build-in semantics proper of each system or community should be interconnected.

This is what the Semantic Web tries to do. It is based on relations between terms, where each term represent a concept. There are semantic relations between terms that capture their semantics. They can be followed when an unknown term is found. There are different types of relations that are used to carry meaning from known terms to unknown ones. From this carried meaning, a partial understanding of the new term can be build. This dynamic and declarative process can be based on a quite simple set of built-in semantic grounds. An example of a semantic network for a set of documents constructed from selected terms is shown Figure 4.3.

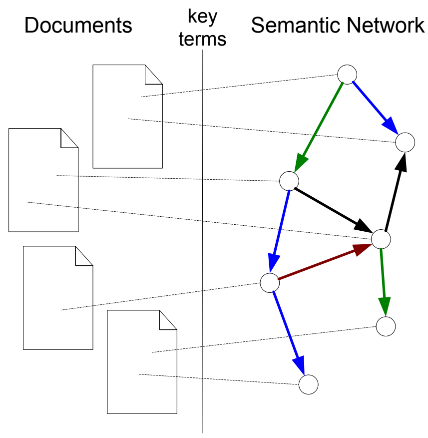

Figure 4.3: Semantic network of terms that situated in the Web becomes a semantic web

The semantic relations emerge from Semantic Web use in a completely free and distributed fashion. This conforms a dynamic agreement, usually partial, of common terms inside a community. Standards are difficult to achieve, and when more and more people is involved, it gets worse. They are so complicated to obtain because standards are a top-down efforts to be applied directly at the global scale. On the other hand, the Semantic Web encourages bottom-up efforts. Global constraints are reduced to the minimum, maintaining local ones, and are used only to focus on a global target.

Additionally, Semantic Web local structures, its documents, are kept simple. However, at the global scale, they build in conjunction a complex system. It provides easier evolvability and thus adaptation to new needs. However, this has a price, there can appear irresolvable questions, contradictions, etc. That is why, commonly, this strategy would not produce complete understanding, only partial.

All the a priori ideas presented in this section are arranged in the Semantic Web principles section 4.7. Then, these principles are put into practice in the Semantic Web architecture section 4.8.

4.5 Semantic Web of Agents

The main intent of the Semantic Web is to give machines much better access to information resources so they can be information intermediaries in support of humans. Thanks to this semantic substrate, and according to the vision described in [13], agents will be pervasive on the Web, carrying out a multitude of everyday tasks.

In order to carry out their required tasks, intelligent agents must communicate and understand meaning. They must advertise their capabilities, and recognize the capabilities of other agents. They must locate meaningful information resources on the Web and combine them in meaningful ways to perform tasks. They need to recognize, interpret, and respond to communication acts from other agents.

As mentioned before, when agents communicate with each other, there needs to be some way to ensure that the meaning of what one agent "says" is accurately conveyed to the other agent. The simplest and most common approach is to build-in the semantics. That is, just assume that all agents are using the same terms to mean the same things.

The assumption could be implicit and informal, or it could be an explicit agreement among all parties to commit to using the same terms in a pre-defined manner, i.e. a standard. This only works, however, when one has full control over what agents exist and what they might communicate.

In reality, agents need to interact in a much wider world, where it cannot be assumed that other agents will use the same terms, or if they do, it cannot be assumed that the terms will mean the same thing. Therefore, we need a way for an agent to discover what another agent means when it communicates. In order for this to happen, agents will need to publicly declare exactly what terms it is using and what they mean. This specification is commonly referred to as the agent's ontology. There is a great interdependence of agent technology and ontologies [56].

Here is where agents and the Semantic Web meet. Agent ontologies define the terms in metadata that describe the resources they work with. Terms in different ontologies are interconnected through the World Wide Web, the space where these agents operate. Therefore, when agents with different ontologies meet they can use connections between their respective ontologies to understand.

4.6 Knowledge Representation

As has been presented in the previous section, the Semantic Web is a quest for a reduced set of commitments upon which understanding can be build. This connects directly with the ideas presented in the Automatic Semantics section 2.7.3. In the Semantic Web, machines deal with the representational dimension of semantics and thus they built a web of concepts connected by relations.

On the other hand, the World Wide Web offers to knowledge representation a promising workspace. Knowledge representation can expand from reduced and constrained application spaces to a huge and continuously evolving one. Moreover, there is the possibility that this complex space would allow knowledge representation techniques build more than isolated human-like intelligent behaviours.

What would emerge might be completely different to what traditionally knowledge representation, more concretely its application by Artificial Intelligence, has envisioned. For instance, there is the vision of an emerging global brain [59, 58] as already mentioned in the Introduction section.

4.7 Semantic Web Principles

The driving force of the Semantic Web is to accommodate the previous objectives and to reuse the existent World Wide Web structure. Web reuse would facilitate a smooth transition from the previous web to the Semantic Web and increase the possibilities of its success.

Therefore, the WWW principles are also considered and some additional ones are included in order to fulfil the augmented requirements. These additional principles are detailed in the next subsections.

4.7.1 Everything identifiable is on the Semantic Web

People, places, and things in the physical world will have online representations identified by Uniform Resource Identifiers which will facilitate effective integration, active participation and be conceptualised in the Semantic Web. URIs are the metadata anchor points to make semantics explicit.

4.7.2 Partial information

Current Web is quite unrestricted; it sacrifices link integrity for scalability. This great lack of restrictions in the Web design make it fundamentally differed from traditional hypertext systems.

This is also a design principle of the Semantic Web ant thus it is also largely unrestricted. Therefore, there should be no fundamental constraint relating what is said, what it is said about, and where it is said. Anyone can say anything about anything. Consequently, it is not expected to have global consistency of all data.

4.7.3 Evolution

Semantic Web provides tools that enable communities resolve ambiguities and clarify inconsistencies. The idea is to use conventions that can expand as human understanding expands.

The Semantic Web must permit distributed communities to work independently to increase the Web of understanding, adding new information without insisting that the old be modified. This approach allows the communities to resolve ambiguities and clarify inconsistencies over time while taking maximum advantage of the wealth of backgrounds and abilities reachable through the Web.

Therefore, the Semantic Web must be based on a facility that can expand as human understanding expands. This facility must be able to capture information that links independent representations of overlapping areas of knowledge.

The Semantic Web encourages the free flow of information but also it should guarantee group boundaries, i.e. restrictions to information access and thus privacy.

4.7.4 Web of trust

All statements found on the Web occur in some context. Applications need this context in order to determine the trustworthiness of the statements. The machinery of the Semantic Web does not assert that all statements found on the Web are "true", i.e. they are propositions that do hold in the world.

Truth, or more pragmatically, trustworthiness, is evaluated by and in the context of each application that processes the information found on the Web. Each application, or agent, states which other agents' statements it does trust. These relations of trust can be partially transitive and dynamically spread trust across the Web. Consequently, a parallel network of statements of trust is then build.

When an application founds a statement, its origin and a trust path to it can be retrieved. From this the statement trustworthiness in the application context can be derived. How this is planed to be done is detailed in the Architecture section 4.8.

4.7.5 Minimalist design

The idea here is to make things simple now that plan for future complexity. The result would be then more than the sum of the parts. This can be summarised in the maxim: "Make the simple this simple, and the complex things possible" or as the Occam's razor principle[2]: "Entities should not be multiplied unnecessarily".

In the Semantic Web it is translated into standardise nor more than is necessary. Construct a global interoperability framework with mapping rules defined over it. They are, in conjunction with the framework primitives, the only global agreement required.

4.7.6 Common Models

To encompass the universe of network-accessible information, the Semantic Web must provide a way of exposing information from different systems. Application or community specific systems may use a variety of internal data models so this implies a requirement for some generic concept of data at a low level that is in common between each system.

Only when the common model is general can any community specific application be mapped onto the model. For instance, databases developed independently are difficult to unify. To solve

[2] http://www.weburbia.com/physics/General/occam.html

this, data can be published using Semantic Web tools and then profit from mapping capabilities to achieve global interoperability.

4.7.7 Rhizome metaphor

One common approach can be seen both in the Web and in the Semantic Web. Indeed, it is the revolutionary idea inherited from the previous Web that is contributed to knowledge representation. It is the philosophy of the rhizome.

The rhizome serves as a metaphor for the multiplicity and infinite interconnectedness of all thought, life, culture, and language. Developed by the theorists Gilles Deleuze and Felix Guattari in their book "A Thousand Plateau's" [25], from which there is an interesting quote:

"A rhizome ceaselessly establishes connections between semiotic chains, organizations of power, and circumstances relative to the arts, sciences, and social struggles. A semiotic chain is like a tuber agglomerating very diverse acts, not only linguistic, but also perceptive, mimetic, gestural, and cognitive: there is no language in itself, nor are there any linguistic universals, only a throng of dialects, patois, slangs, and specialized languages. There is no ideal speaker-listener, any more than there is a homogeneous linguistic community.... There is no mother tongue, only a power takeover by a dominant language within a political multiplicity. Language stabilizes around a parish, a bishopric, a capital. It forms a bulb. It evolves by subterranean stems and flows, along river valleys or train tracks; it spreads like a patch of oil. It is always possible to break a language down into internal structural elements, an undertaking not fundamentally different from a search for roots. There is always something genealogical about a tree. It is not a method for the people. A method of the rhizome type, on the contrary, can analyse language only by decentring it onto other dimensions and other registers. A language is never closed upon itself, except as a function of impotence."

Gilles Deleuze and Felix Guattari, "A Thousand Plateau's"

The Web provided the rhizome approach to the information level, where the rhizome approach stands for a hierarchy less, open and decentralised way of organisation. This approach, applied to information, has showed as the best suited in an Internet-connected world. Therefore, the novelty, and the challenge, is to apply it to the knowledge level, i.e. constructing a Web of interrelated ontologies.

4.8 Semantic Web Architecture

The previous ideas and principles to complete the Web are being put into practice under the guidance of the World Wide Web Consortium. To reduce the amount of standardisation required and increase reuse, the Semantic Web technologies have been arranged into a layer cake shown in Figure 4.4. The two base layers are inherited from the previous Web. The rest of the layers try to build the Semantic Web. The top one adds trust to complete a Semantic Web of trust.

The Semantic Web layers are arranged following an increasing level of complexity from bottom to top. Higher layers functionality depends on lower ones. This design approach facilitates scalability and encourages using the simpler tools for the purpose at hand. All the layers are detailed in the next subsections.

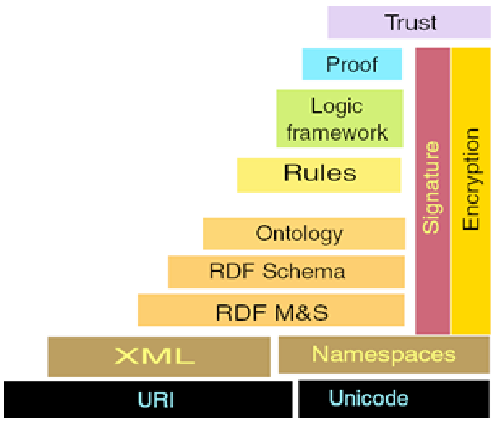

Figure 4.4: Semantic Web Stack, from Tim Berners-Lee presentation for Japan Prize, 2002

4.8.1 URI and UNICODE

The two technologies that conform this layer are directly taken from the World Wide Web. URI provides global identifiers and UNICODE is a character-encoding standard that supports international characters.

In few words, this layer provides the global perspective, already present in the WWW, for the Semantic Web.

4.8.2 XML and Namespaces

The Semantic Web should smoothly integrate with the Web. Therefore, it must be interwoven with Web documents. HTML is not enough to capture all that is going to be expressible in the Semantic Web. XML is a superset of HTML that can be used the serialisation syntax for the Semantic Web. XML was initially tried but more recently other possibilities have been developed. They are presented and compared in the next section.

Namespaces where added to XML to increase its modularisation and the reuse of XML vocabularies in conjunction with XML Schemas. They are also used in the Semantic Web for the same purpose.

4.8.3 RDF Model and Syntax

The RDF Model and Syntax specification [7] defines the building blocks to realise the Semantic Web. This is the first layer that was specifically developed for it. This specification defines the RDF graph model and the RDF abstract syntax.

The RDF graph model defines a structure composed of nodes and directed edges between nodes. The structure of nodes and edges conform directed graphs that model the network of terms and relations between terms of the Semantic Web. The nodes and relations are called resources and are identified by URIs. Each node has its own URI and there are different types of relations that also have an URI, they are called properties. Figure 4.5 shows and example of RDF graph model.

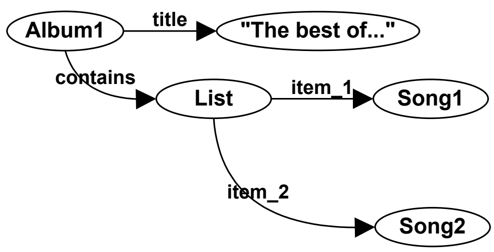

Figure 4.5: RDF Graph Model example

Particular edges are identified by the triad composed by the origin node, the property and the destination node. Triads are called triples ore RDF statements and they are the RDF abstract syntax. Graphs can be serialised as a set of triples, one for each edge in the graph. Both representations are equivalent so the graph model can be reconstructed from the set of triples.

Triples can also be assigned an explicit identifier, i.e. an URI. This process is called reification. A new node is created that represents the triple and it is associated to three nodes for the three triple components. The origin node is associated using the "subject" property, the property with the "predicate" property and the destination node with the "object" property. Reification is useful to say things about RDF statements. For an example of use, see Figure 4.6.

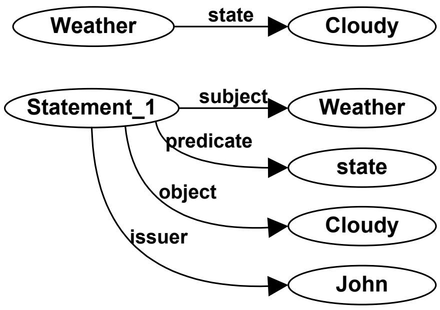

Figure 4.6: Triple reification example

Abstract triples are the common model to which diverse data structures can be mapped. For instance, relational tables can be translated to a set of triples. Notwithstanding, triples are abstract entities. They are realised for communication using serialisation syntaxes.

The XML syntax has already been introduced in the previous section, it facilitates integrating Semantic Web documents in the current HTML/XML web. The other possibilities are N-Triples and Notation 3 syntax, http://www.w3.org/DesignIssues/Notation3.html. The former is the nearest to the abstract form, a series of triples with subject, predicate and object identified by their URI. The latter uses many syntactic tricks to improve human readability and make serialisations more compact. It is the more human aware syntax and, like XML serialisation, it uses namespaces for modularisation.

4.8.4 RDF Schema

Simple RDF provides the tools to construct semantic networks. They are a knowledge representation technology presented in the Semantic Networks section 2.10.3. Nonetheless, there is still a lack of many semantic network facilities not available with RDF.

There are no defined taxonomical relations. They are defined in the RDF Schema specification [17]. Taxonomical relations leverage RDF to a knowledge representation language with capabilities similar to semantic networks. This enables taxonomical reasoning about the resources and the properties that relate them.

RDF Schema specification provides some primitives from semantic networks to define metadata vocabularies. RDF Schemas implement metadata vocabularies in a modular way, like XML Schemas. Schema primitives are also similar to Object Orientation constructs they also evolved from the semantic networks tradition. The more relevant ones are detailed next and an example of their use is shown in Figure 4.7:

- **type**: it is a property that relates a resource to a Class to which it pertains. The resource is categorised as a member of this Class and thus it possesses its characteristics.

- **Class**: it is a set of things that share some characteristics; they have a common conceptual abstraction. A class models the concepts present at the referential semantic level. This subject was introduced in the Sense and Reference section 2.7.2.

- **subClassOf**: this property holds the taxonomical relations between classes. If class B is a subclass of class A, then class B has all the typical characteristics of class A plus some specific ones that can distinguish it from A.
  For instance, if a RDF graph states that, a resource R is a "Mammal", i.e. R has type the class "Mammal", and that "Mammal" is subclass of "Animal"; then it can be deduced by taxonomical reasoning that R is also an "Animal".

- **subPropertyOf**: this property creates the taxonomy of properties. If property B is a subproperty of property A, then whenever it is stated that the property B holds between two resources it can be deduced that A also holds.
  For instance, if a RDF graph says that, a resource R is related to another one S through a relation called "motherOf" and "motherOf" is a subproperty of "parentOf"; then it can be deduced that the property "parentOf" also holds between resources R and S.

- **domain, range**: Both are properties that associate other properties to classes. They constraint the classes to which the associated properties can be connected. Domain defines all classes to which the subject resource of the triples where property appears must belong. The same is applicable for range but constraining the object resource.

A first simile can be established at this stage. While HTML makes the Web behave like a global book when viewed at the worldwide level, RDF Model and Syntax plus RDF Schema make it behave like a global database.

Another simile with XML can also be clarifying. The basic RDF primitive, the graph, can be compared with the XML one, the tree. However, as an XML tree, an RDF graph is on its own

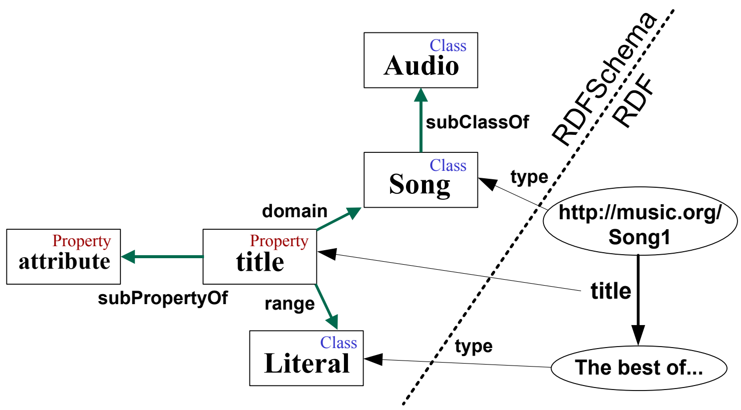

Figure 4.7: Example of RDF Schema

basically unrestricted. Therefore, in order to capture the semantics of a particular domain, some primitives to build concrete "how things are connected" restrictions are necessary. RDF Schema provides these restriction-building primitives. It can be compared to XML Schema or DTDs, which provide building blocks to define restrictions about how XML elements and attributes are related.

4.8.5 Ontology

Ontologies are necessary when the expressiveness achieved with semantic network-like tools is not enough. Metadata vocabularies defined by RDF Schemas can be considered simplified ontologies. The tools included in this layer rise the developed vocabularies to the category of ontologies. For a comparative with XML Schemas, see Table 4.1.

Ontologies, which were defined in the Knowledge Representation Ontology section 4.8.5, are specially suited to formalise domain specific knowledge. Once it is formalised, it can be easily interconnected with other formalisations. This facilitates the interoperability among independent communities and thus ontologies are one of the fundamental building blocks of the Semantic Web.

Description Logics are particularly suited for ontology creation. They were introduced in the corresponding Knowledge Representation subsection 2.7. The World Wide Web Consortium is currently developing a language for web ontologies, OWL [23]. It is based on Description Logics and expressible in RDF so it integrates smoothly in the current Semantic Web initiative.

Description Logic makes possible to develop ontologies that are more expressible than RDF Schemas. Moreover, the particular computational properties of description logics reasoners make possible efficient classification and subsumption inferences.

When comparing ontologies and XML schemas directly we run the risk of trying to compare two incomparable things. Ontologies are domain models and XML schemas define document structures. Still, when applying ontologies to on-line information sources their relationship becomes closer. Then, ontologies provide a structure and vocabulary to describe the semantics of information contained in these documents. The purpose of XML schemas is prescribing the structure and valid content of documents, but, as a side effect, they also provide a shared vocabulary for the users of a specific XML application. Differences between ontologies and schema definitions: • A language for defining ontologies is syntactically and semantically richer than common approaches for databases. • The information that is described by an ontology consists of semi-structured natural language texts and not tabular information. • An ontology must be a shared and consensual terminology because it is used for information sharing and exchange. • An ontology provides a domain theory and not the structure of a data container

Table 4.1: XML Schemas vs. Ontologies

4.8.6 Rules

The rules layer allows proof without full logic machinery. Similar rules are those used by the production systems presented in the corresponding Knowledge Representation subsection 2.10.2. They capture dynamic knowledge as a set of conditions that must be fulfilled in order to achieve the set of consequences of the rule.

The Semantic Web technology for this layer is the Semantic Web Rule Language (SWRL) [61]. It is based on a previous initiative called Rule Modelling Language (RuleML) [55]. As RuleML, SWRL covers the entire rule spectrum, from derivation and transformation rules to reaction rules. It can thus specify queries and inferences in Web ontologies, mappings between Web ontologies, and dynamic Web behaviours of workflows, services, and agents.

4.8.7 Logic

The purpose of this layer is to provide the features of FOL. First Order Logic was described as the most significant type of logic in the Logic types section 2.6.1. With FOL support, the Semantic Web has all the capabilities of logic available at a reasonable computation cost as shown in the Deduction section 2.8.1.

There are some initiatives in this layer. One of the first alternatives was RDFLogic [12]. It provides some extensions to basic RDF to represent important FOL constructs, for instance the universal ($\forall$) and existential ($\exists$) quantifiers. These extensions are supported by the CWM [11] inference engine. Another more recent initiative is SWRL FOL [89], an extension of the rule language SWRL in order to cope with FOL features.

4.8.8 Proof

The use of inference engines in the Semantic Web makes it open, contrary to computer programs that apply the black-box principle. An inference engine can be asked why it has arrived to a conclusion, i.e. it gives proofs of their conclusions.

There is also another important motivation for proofs. Inference engines problems are open questions that may require great or even infinite answer time. This is worse as the reasoning medium moves from simple taxonomical knowledge to full FOL. When possible, this problem can be reduced by providing reasoning engines pre-build demonstrations, proofs, that can be easily checked.

Therefore, the idea is to write down the proofs when the problem is faced and it is easier to solve as the reasoning context is more constrained. Further, proofs are used whenever the problem is newly faced as a clue that facilitates reasoning on a wider content.

Many inference engines specialised in particular subsets of logic have been presented so far. For instance:

- Prolog for logic programming.
- The production system Jess .
- The FaCT implementation of Description Logics reasoners.
- The CWM inference engine presented in the previous section.

4.8.9 Trust

This is the top layer of the Semantic Web architecture. Agents that want to work with the full-featured Semantic Web will be placed over it. They will conform the Web of Trust.

The trust layer makes use of all the Semantic Web layers below. However, they do not provide the required functionality to trustily bind statements with their responsible parts. This is achieved with some additional technologies that are shown in the right part of the Semantic Web stack Figure 4.4.

The used tools are digital signature and encryption. Thus, the trust web will make intensive use of Public Key Infrastructures. They are already present in the Web, for instance as digital certificates identifying parties that sign digital contracts. Notwithstanding, there is not a widespread use of them.

The premise is that Public Key Infrastructure is not of extended use because it is not a decentralised web structure. It is hierarchical and therefore rigid. What the Semantic Web might contribute here is a less constraining substrate of use. The web of trust is based on the graph structure of the Web. Moreover, it supports the dynamic construction of this graph. These features might enable the common use of Public Key Infrastructure in the future Web.

To conclude, the final Semantic Web picture contains reasoning engines complemented with digital signatures to construct trust-engines. Then, a Trust Web can be developed with rules about which signed assertions are trusted depending on signer.

4.9 Semantic Web Services

The Semantic Web initiative started before the Web Services one, which was presented in the corresponding Web Technologies sub section 3.4.2. Web services are based on UDDI, WSDL, and SOAP technologies, which provide limited support in mechanising service recognition, configuration, comparison, combination and automated negotiation. Nonetheless, the idea behind Web Services is to realize complex workflows and business logics.

Therefore, some Semantic Web and Web Services initiatives are converging. There are development like OWL-S[81] that employ Semantic Web technologies for services descriptions. OWL-S models web services as sets of semantically defined processes. Based on the described service semantics, mediation is applied based on data and process ontologies and the automatic translation of their concepts into each other. The top-level part of the OWL-S process ontology is shown in Figure 4.8.

Technologies from both initiatives are complementary. Web services semantic description languages like DAML-S are situated as a higher-level layer above WSDL. UDDI, WSDL and SOAP continue to be used in semantic web services as the implementation machinery. DAML-S is used when advanced processing is needed to perform service recognition, configuration, comparison, combination and negotiation. A mapping between both is shown in Figure 4.9.

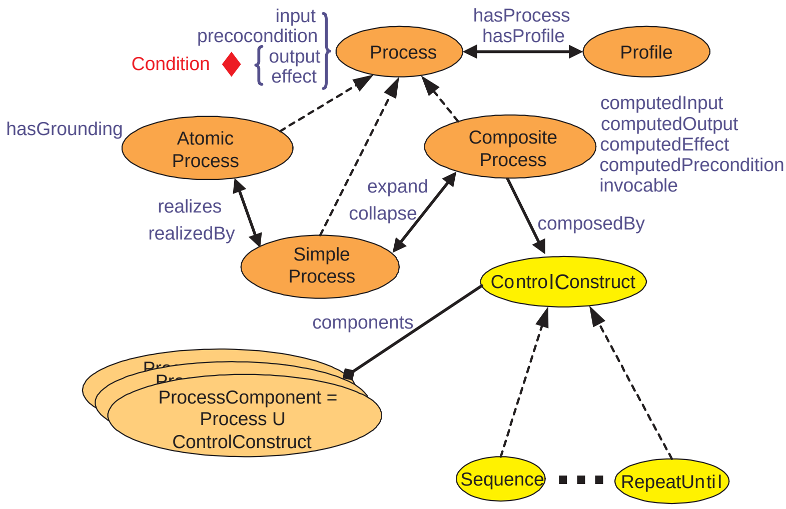

Figure 4.8: Top level of DAML-S process ontology

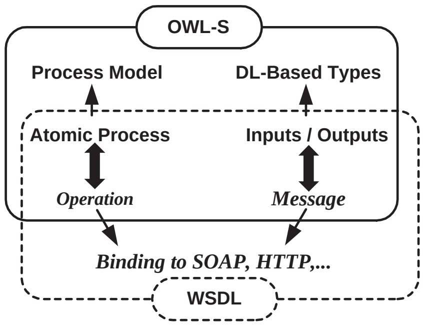

Figure 4.9: OWL-S to WSDL mapping

Chapter 5

Rights Expression Languages

5.1 Introduction

Lately, there have been great changes in the copyright market motivated by the digital and Internet revolutions. First, these revolutions have introduced new risks in the classical market, which was basically based on the distribution of physical instances of content. Second, they have opened opportunities to create new markets based on digital creations and the Internet distribution medium.

In order to manage this new situation, the main approach is to take profit from the new technological opportunities in order to develop systems to manage and protect digital works. This is referred to as Digital Rights Management, or DRM. DRM is a system of IT components and services along with corresponding law, policies and business models which strive to distribute and control content and its rights.

There is no universal DRM system (DRMS), only implementations of DRMS that satisfy the needs of specific value chain users. It is impossible to standardise functions performed in existing value-chains (we do not know how today's value-chains will evolve) and future value-chains (we do not know what they will be). It is possible to standardise lower-level functions, i.e. primitives, between value-chain players. Then current value-chain function can be implemented as a combination of the primitives. Moreover, in the future, it would be easier to develop new functions combining the existing primitives in almost all cases. This approach supports the possibility to inject continual innovation in the system.

5.1.1 Motivation

With a digital creation, it is possible to make a copy almost instantly, it will cost nothing and the end result will be a perfect copy of the original. The digital file is protected by the very same copyright law that a hard copy is, the one that does not really prevent us from making copies. Because law is not working as a preventive measure, there is some justification that only a technology-based protection will ever work to protect digital works.

One of these technologies is encryption. It does not prevent you from copying a file. The protection that encryption provides has nothing to do with copying; instead, encryption prevents access to the content of a file. Encryption does not provide any control over copying.

Using encryption, DRM systems tie a digital file to a particular piece of hardware. This has obvious problems when there are hardware changes or users trying to enjoy their content in multiple devices. A better solution is to connect the digital file directly to the person that can enjoy it. This approach requires current under work technologies called "trusted systems".

The controls over access to the file are just the first step in DRM. There are many more controls that can be applied, such as controls over whether you can print from the file, copy passages to the clipboard, or whether the file expires after some time period. One example of DRM system that allows such kind of usage controls is Adobe Reader 6.0.

However, in many cases, these controls may become too hard and even contrary to copyright. For instance, text-to-speech facilities for accessibility purposes may be disabled or the print option unavailable even for public domain document.

The controls in current DRMS, i.e. basically e-book readers, are quite basic. But the future of digital rights management is in the development of something called a Rights Expression Language (REL). A REL will allow a publisher to designate a much more detailed and complex set of usage controls. The REL will be able to control the number of times a text can be read, can set timed controls (i.e., "for two weeks starting today"), and can manage complex relationships of distribution, sale and lending.

A REL is a language that expresses the rights you have in relation to a file. This language differs from legal language because it must be a formal language that can be interpreted unambiguously by computers. RELs are computer-oriented so the kind of controls they can model must be of the kind computers can manage, i.e. quantitative controls: time, unit-based dimensions (e.g. pages, chapters...), money, etc. More details about REL and some of the main REL developments are given in the next sections.

5.1.2 Trusted Systems

Rights expression languages just express rights statements; they have no means to enforce them. In order to enforce the statements, the REL expressions must be interpreted in the context of a system. The system interacts with all the involved parties and must be secure from end to end, i.e. it must be a "trusted system".

In order to trustily manage rights transaction many individual systems must be coordinated. First, there is the end-user computer. In order to make this computer trustworthy the whole system must be able to trust that the end-user computer will obey the rules of the rights management software, regardless of what the user does with it. For instance, if the user has been granted the use of a digital resource for a limited period of time, a trusted DRMS must make it impossible for the user to alter the date of his computer in order to consume the resource outside the granted time period. This is intended to prevent piracy, because a program that would break the encryption on a protected file simply would not be allowed to run.

5.1.3 DRM and the Law

With DRM in place, although copyright law will still recognize users' rights to fair use, they may not be able to exercise them. Currently, DRM is not conceived as an implementation of copyright law. It is intended as a system for the protection of digital works and DRM will implement licenses rights or grants like controls that can be expressed in a computer environment. Another important difference between DRM and copyright law: copyright law does not attempt to anticipate every possible use of a copyrighted work.

A digital rights management system functions in exactly the opposite way. Where copyright law is an expression of "everything that is not forbidden is permitted", DRM takes the approach of "everything that is not permitted is forbidden". This is seen as a necessary requirement to create secure software but this may limit legal use, which might create a very negative view of DRM in those traditionally enjoying this exceptional uses. Moreover, it might have great implications for future uses of protected works. If the starting point for DRM is that something not explicitly authorised is forbidden, DRM systems will automatically forbid unforeseen ways of using content.

5.2 Rights Expression Languages

As it has been already introduced, Rights expression languages (RELs) are part of the technology of digital rights management. As any language, their objective is to be a vehicle of expression. This generic goal is concretised in the DRM field as follows:

- The expression of copyright
- The expression of contract or license agreements

Moreover, there is a clear purpose of this expressions, which is what relates RELs to DRM:

- The control over access and/or use

The degree to which RELs are intended to be machine-actionable is a determinant in the kinds of rights that can be expressed in the REL. A machine-actionable REL must use very precise language and can nearly guarantee compliance with the terms of the machine-readable license. This REL cannot, however, support social or legal concepts like "fair use". On the other hand, broader and less precise RELs must rely on agreement and trust for enforcement, which means that there is a risk of unauthorized use. In the RELs Analysis section 5.2.3 we will analyse three of the main RELs: Creative Commons [78], MPEG-21 REL [120] and ODRL [64]. Now, they are situated in relation with these objectives in Figure 5.1.

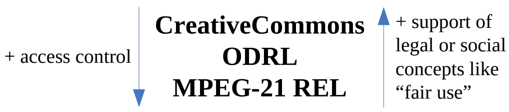

Figure 5.1: Comparing RELs functionalities

Creative Commons functions specifically in the open access environment of the World Wide Web. ODRL is a general-purpose language that allows, but does not require, some actionable control over resource use. MPEG-21 REL is a general language that is formally described and fully actionable within a trusted systems environment.

For practical use it is expected that organizations should have to tailor the chosen REL to their particular needs. However, any one of these may provide a suitable basis for that development. In particular the two general-purpose languages, ODRL and MPEG-21 REL have a rich vocabulary that can be reduced or expanded to create a REL for a specific purpose.

Copyright law is the default agreement that exists when no other arrangement has been made between parties. A contract is a stated agreement between any two parties. Therefore, all RELs have some relationship to copyright law because it exists as a default environment. Most, however, make little reference to law.

Copyright law makes a statement about ownership of intellectual works and the rights of various parties. It gives particular rights to the copyright owner over a limited set of actions: reproduce the work, derive other works from the work, distribute copies of the work, perform the work, and display or perform the work publicly. Copyright law does not make specific reference to using materials, such as viewing or listening; these are considered to be "normal use" and assumed to be permitted.

The approach of almost all REL initiatives is that there is little that a rights expression language can or should do in relation to the copyright law. Moreover, rights expression languages that are intended to be machine-actionable are expressly not intended to implement copyright law. Some early attempts to use RELs to express legal concepts like "fair use" did not succeed. The copyright law, although carefully worded, simply cannot be expressed in the kind of languages that is used to model RELs.

This is especially true of the key concept of "fair use". Fair use is a deliberately vague exception to the monopoly rights of the copyright holder. It says essentially that although the copyright holder has the exclusive right to make copies of the work, members of the public can also make copies if their use is "fair". There is no *a priori* test for whether a use is fair. Electronic systems need an unambiguous and quantitative definition that they can act on, and the copyright law does not provide that.

MPEG-21 REL and ODRL are focused on the parties to the license. Both refer to the issuer of the license, but have no reference to copyright. On the other side, Creative Commons is more concerned with copyright. The machine-readable statements keep room for copyright holder's names and contact information. Moreover, the human-readable version of the statements can be augmented to a fully legal document, although it is not machine-readable.

Finally, for the implementation of controls goal, it is important to state the difference between *agreement* (contract) and *execution* (control). A contract is essentially an agreement to behave in a certain manner. Control is an actual implementation of the rights and limitations. When there is a controlling mechanism in place the parties are unable to violate the terms of an agreement even if they should wish to. The same language that expresses contracts may be used in control mechanisms if it is designed in such a way that it can be implemented in software or hardware.

5.2.1 History

Mark Stefik developed the first Rights Expression Language (REL) for Digital Rights Management at Xerox PARC in the early 1990's. The motivation was the need for protection for digital materials in order to support online commerce. Then, a trusted systems environment could be developed that would provide the level of security needed to allow digital commerce to flourish [107].

That system would need a machine-readable language to represent rights statements. The first REL by Stefik was called the Digital Property Rights Language (DPRL). In 1998, it was licensed to a company founded by Microsoft and Xerox called ContentGuard. Then, its development continued and it was renamed eXtensible Rights Markup Language (XrML) in 2001 [121]. In 2003, the MPEG-21 Standard [22] work started with a part of it dedicated to a rights expression language and a complementary rights data dictionary. The proposed REL is based on the last version of XrML.

Meanwhile, there were others also working in this arena. Renato Ianella of IPR Labs proposed the Open Digital Rights Language (ODRL) in 2000 as an open standard rights language. ODRL has been adopted by the Open Mobile Alliance as its rights expression language in the mobile phones domain. Moreover, the concept of rights expression languages has been appearing in almost any metadata initiative for digital resources, e.g. ONIX, OAI, METS, Dublin Core, MARC and others.

RELs are just languages to express the statement that have to be interpreted and implemented by the rights management systems; they do not act directly on content. Currently, there are some more or less mature REL initiatives but the corresponding rights management systems are still at an early stage of development. Moreover, the development is slowed by the great complexity of RELs, which makes them difficult to implement.

The main contenders for a generalized REL today appear to be MPEG-21 REL (based on XrML) and ODRL. Another of the main RELs, first developed in 2002, is Creative Commons. It seems the best alternative for open environments because it does not impose any access control. Although this lack of control mechanisms, it has great acceptance in the Web and it is being used to share contents on a free way, e.g. educational contents, free software, etc.

We have highlighted three of the main RELs but there are more. And all are trying to provide an universal language for rights expression. The consequence is that, currently, the intuition is that there is not and probably will never be a universal REL. Each one fits quite particular requirements and, as digital rights management evolves and gets more common, even more specialised requirements would open more REL development possibilities.

5.2.2 Using Rights Expression Languages

The RELs that have been considered are quite different in terms of how they themselves can be licensed and used. ContentGuard does not require a license but the company holds numerous patents on digital rights management technologies. Their patents may cover more than just XrML; in particular one patent titled "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) may be interpreted to cover all rights expression languages. Although ContentGuard has not yet pressed its patent with others developing and using RELs, they have stated that their position is that the patent provides that capability.

ODRL is explicitly a license and patent-free technology that has been developed in the spirit of open source technology. The same applies for Creative Commons, which models its licenses on those of the open source movement, and has no license requirements for making use of the CC materials.

5.2.3 RELs Analysis

In order to make a survey of the current REL state of the art, we have performed an analysis over the main RELs: Creative Commons, ODRL and MPEG-21 REL. First of all we are going to detail the different aspects considered in this analysis: how do they represent contract, how do they manage control, their general architecture and finally the data elements that conform the language. Then, each of these RELs is analysed respect to these aspects. Finally, other REL initiatives are introduced shortly. This analysis has been largely inspired by Karen Coyle's Report on Rights Expressions Languages for the Library of Congress [20].

Contract

Rights or permissions beyond those included in copyright law are covered by contract or license. A copyright holder can extend copy and distribution rights through the mechanism of contracts and licenses. These agreements can give more or fewer rights to the users of the copyrighted material than would be covered by copyright law. Contracts are agreements between individuals, institutions, or groups and do not apply to the public at large. They can contain any constraints that the parties agree on.

In the RELs domain generally agreements are referred as licenses. This is indicative of the common view that in copyright contracts one party is giving specific permissions to another, rather than a general contractual agreement between parties.

A contract language typically considers some of the following:

- Named parties and their roles (can be middlemen, i.e. retailers).
- Statements of access and usage that go beyond pure copyright law.
- Exchange of value (payments) for services or actions.

In general, RELs expressions are statements about privileges granted by one party to another.

Control

Neither copyright law nor contracts assert any actual control over the behaviour of users of materials. They rely on the parties to act within the stated agreement or law. Because digital materials must be mediated through software and hardware, it is possible to exercise *a priori* control over access to and use of the content through that technology.

The functions of control and contract tend to have data elements in common because they both represent license terms. Control is distinctive because it is designed to be machine-enforceable, therefore it will use a highly formalized expression.

In order to exercise control there are two alternatives. The simplest one is to use dedicated devices that facilitate the implementation of the controls as their functionality is already constrained by their hardware. On the contrary, the second alternative is based on general-purpose computers. They constitute an open framework that allows users to perform a great amount of actions that should be controlled. This makes the implementation of control much more difficult. The approach in this case is to develop trusted systems, i.e. safe areas within our general computing environment where such controls can be implemented. In trusted systems, certain functions available to other software will be under the control of the trusted system and not the computer user.

There are two key points where control can be exercised, the resource *access* and the resource *use*. Access controls limit who can receive or download a file. Access control is usually not covered by RELs as, for this task, they rely on external systems or the computation environment where they are implemented, e.g. the operating system or a public key infrastructure.

Usage controls determine what a user can do once the digital resource has been obtained. Since digital resources must be rendered in some way to make them human-perceivable, usage controls are generally built into the software and/or hardware that enable that perception. Access and usage controls can work together or separately.

Control-oriented languages, e.g. MPEG-21 REL or ODRL, express their controls as being license-based. The license is not a contract in the legal sense but it is a digital rendering of permissions for use. These permissions might also be expressed in a human-readable contract that has legal ramifications, but that is assumed to be outside of the REL itself. For these controls to function within an automated system, every permitted usage type must be explicitly granted in order for the rendering software to securely protect the resource. Therefore, a fully functioning rights language designed for automated control must define every possible allowable usage.

Data Elements

RELs are made up of data elements that express the rights situation. In general, a rights system is made up of *resources*, *agents* that interact with those resources, sets of *rights* or permissions, *constraints* on those rights, and *requirements* such as payments.

Agents This element is used to identify the party or parties involved in the expressed rights statement. It is quite general so it can represent different roles in the environment of the REL.

Resources This element is basically used to refer to the resource identifier. It is assumed that further details lay outside the REL and are linked in some way, usually by the identifier.

Rights Rights are the core of any rights expression language. RELs use different terms to refer to them. Some talk about permissions, others about grants, etc. However, the purpose is always the same, to express a set of allowed actions over the resource. Rights refer to end-user actions, such as playing or printing, and also to other parties' actions, such as copying or communicating to the public.

The list of allowable actions varies greatly between RELs. The general purpose ones, which try to cover the widest range of situations, have the most extensive lists of actions. Moreover, some of them are designed for expansion of their rights metadata elements. Others are very specific to an application domain and they just cover the kinds of actions that are relevant to such domain.

Rights can generally be assigned to one of these types: manage, re-use, transfer and use:

- **Manage**: The manage type of actions covers the housekeeping tasks like install or backup. This kind of rights applies to the content package, not to the intellectual content that is contained in that package.

- **Re-use**: There is a re-use when all or part of a resource can be modified or incorporated into other resources. For instance, audio files are re-used when they are sampled; textual materials are re-used when a passage is quoted. These rights are difficult to manage in the context of a rights enforcement system because this kind of actions usually are usually performed over resources different from the one protected by the technology, they have just an intellectual relation that is difficult to derive using computerised systems. Examples of these technology controls are to prevent changes to a file or to block copying portions of a file to the system's clipboard.

- **Transfer**: The transfer category of rights applies to the actions that allow a person or agent to transfer some specific rights to another person or agent. For instance, to sell or to lend are transfer actions.

- **Use**: This is the kind action that affects end-users of content. They are related to how the consumer will experience the content. Use rights include rendering actions like to play, to display or to print.

These categories are detailed next and there is a summary of all the right elements defined by the considered RELs in the RELs Overview section 5.6.

Constraints Permitted actions are modified with constraints. They are based on any criteria that can logically be applied to the action, but tend to be quantitative elements in actionable RELs, i.e. time, payment or units. There are also constraints on aspects about the user and the use that are not quantitative. For instance, quantitative constraints based on units (page, chapter...), time (dates, duration...) and place (territory, region...) can be applied to a right like "play" in order to build a license like "the first chapter of this DVD can be played in the USA and Canada region 100 times each year from January 1, 2005 to December 31, 2010".

Conditions They are the specific requirements that must be fulfilled before the user can exercise the rights. The most common is payment, for instance to give a credit card number in order to complete a song acquisition process. Some of these conditions can be enforced, e.g. checking the credit card number, but others not. For instance, a usage condition like "attribution" cannot be enforced by a digital rights management system because the required action lies outside the scope of the DRMS and thus it cannot test if it is fulfilled.

5.3 Creative Commons (CC)

Creative Commons (CC[1]) provides a legal framework and expression language for building a "some rights reserved" medium for resources sharing on the Web. The resources can be audio, images, text, video, etc. The expression language allows defining CC licenses that are machine-readable. However, there is no machine-actionable control over use of the content that carries such a license. This means that, for automation purposes, such licenses can be used for cataloguing purposes but not for DRM.

CC relies on existing copyright law to protect digital content. In order to connect CC licenses to copyright law, the Creative Commons initiative has created a set of human-readable "classical" licenses that legally define the predefined set of CC licenses. These licenses are also available summarised for users not interested in the legal details and, as it has been said in the previous paragraph, in machine-readable form. The philosophy of the Creative Commons licenses is similar to the open licensing scheme of the Free Software Foundation, the GNU General Public License.

The Creative Commons REL is based on RDF metadata, see the RDF Model and Syntax section 4.8.3, and thus it is defined by a RDF Schema, see the RDF Schema section 4.8.4. The CC licenses in their machine-readable form have two parts: *Work* and *License*, which are classes in the RDF Schema terminology. The *Work* section describes the resource to which the license pertains using simple Dublin Core [24] metadata elements. The *License* part is more specific, it defines the concrete actions that are required, permitted or prohibited by the license referred by the *Work* part. More details about the Creative Commons rights expression language are given in the following subsections.

5.3.1 Contract

Creative Commons licenses are centred on these three axis:

- Require attribution or not.
- Disallow derivatives (modifications) or not.
- Disallow commercial use or not.

The objective of the Creative Commons project is to promote creativity by building a great base of material that can be re-used for new creations. This re-usable creations base is build using the

[1] http://www.creativecommons.org

Internet infrastructure where the materials are shared and related to a rights expression. These expression use the CC rights language that was designed to support the re-use of material. Therefore, all of their license statements address the issue of re-use.

In any case, even if either modifications or commercial use are disallowed, this does not mean that those actions are strictly forbidden. This is a less strict interpretation of "disallow" than it is common in most RELs. CC contracts define "disallow" to mean that such use must be negotiated with the copyright holder.

As it has been said, CC licenses are expressed using a machine-readable language, the CC REL but behind this simple CC license there is a fully expressed license using legal terminology. The REL acts as a simplified outline that can be used for automated cataloguing of content based on rights terms, but the rights language is considered only a summary of a human-readable license. The expanded version also explains the uses and requirements that are covered by copyright law, since these are assumed to apply to all materials on the Internet, whether or not a CC license has been assigned.

5.3.2 Control

Creative Commons is not concerned with access control because its application domain is based on web-accessible documents. Moreover, it is not concerned with usage control either because its objective is to promote the re-use of these resources and not to hinder it. Therefore, Creative Commons does not have any access or usage control support and it entirely relies on the Copyright law framework. This makes CC the only one that does not have control mechanisms among all the studied RELs. On the other hand, it is the one that best supports copyright law, specially the exceptions like fair use or private copy. This is because CC is not obliged to make the assumption that everything that is not allowed by a license is automatically forbidden, on the contrary to control-oriented RELs.

5.3.3 Data Elements

The data elements defined by the Creative Commons REL are specified in the following subsections depending on their category as specified in the RELs Analysis section 5.2.3. There is an example of CC REL document in Table 5.1.

```
<rdf:RDF xmlns="http://web.resource.org/cc/"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<Work rdf:about="http://example.org/gnomophone.mp3">
<dc:title>Compilers in the Key of C</dc:title>
<dc:description>A lovely classical work on compiling code.</dc:description>
<dc:creator><Agent><dc:title>Yo-Yo Dyne</dc:title></Agent></dc:creator>
<dc:rights><Agent><dc:title>Gnomophone</dc:title></Agent></dc:rights>
<dc:date>1842</dc:date>
<dc:format>audio/mpeg</dc:format>
<dc:type rdf:resource="http://purl.org/dc/dcmitype/Sound" />
<dc:source rdf:resource="http://example.net/gnomovision.mov" />
<license rdf:resource="http://creativecommons.org/licenses/by-nc-nd/2.0/" />
<license rdf:resource="http://www.eff.org/IP/Open_licenses/eff_oal.html" />
</Work>
<License rdf:about="http://creativecommons.org/licenses/by-nc-nd/2.0/">
<permits rdf:resource="http://web.resource.org/cc/Reproduction" />
<permits rdf:resource="http://web.resource.org/cc/Distribution" />
<requires rdf:resource="http://web.resource.org/cc/Notice" />
<requires rdf:resource="http://web.resource.org/cc/Attribution" />
<prohibits rdf:resource="http://web.resource.org/cc/CommercialUse" />
</License>
</rdf:RDF>
```

Table 5.1: Creative Commons license example

Agents

For the element used to identify the parties involved in rights expressions, Creative Commons defines the element *Agent* which defines as "people or things that do stuff". The *Agent* element just identifies the corresponding party and it does not specify its role. More details about the party can be specified using other metadata schemas, e.g. the Dublin Core. With Dublin Core it is possible to state the agent's name using the *title* element and different roles using the following elements:

- *creator*: the agent who created the resource.
- *publisher*: the agent responsible for making the resource available.
- *contributor*: the agent who contributed to the creation of the resource.
- *rights*: the agent who holds the copyright on the resource.

There are no end-users identified in CC because it operates in an open Web environment. No specific end users are identified as it is directed to any user that accesses the resource.

Resources

Differently to almost all RELs, CC does not rely on external metadata and enables a way to include more than just the resource identifier in order to describe it. CC includes the *work* element that allows for the use of Dublin Core data elements to describe the resource. Examples of the common Dublin Core elements used in CC are: title, description, creator's name, copyright holder's name, date, etc.

Rights

The *work* part of CC licenses has a *license* element that points to the license governing the described resource. The rights that are permitted or prohibited are specified inside the *License* element (note the leading capital letter). The available rights are classified and detailed in the following subsections.

Manage Creative Commons does not consider maintenance tasks such as the installation or backup of the files that usually contain digital creations. This is because CC centres on Copyright law and creations and it does not take into account rights about the resource package, only about the intellectual content.

Re-use ***DerivativeWorks***: this element is related to the right to create derivative works from a resource. If this element appears related to a *permits* element, the corresponding license specifies that derivations from the governed resource may be created and reproduced. On the other hand, if it appears related to a *prohibits* elements, then no derivations can be produced.

Transfer ***Distribution***: when this element is permitted, the corresponding CC license specifies that the work (and, if authorized, derivative works) may be distributed, publicly displayed, and publicly performed.

Use ***Reproduction***: this right is related to the reproduction of the work, i.e. to make copies of it. It can be permitted or prohibited by a Creative Commons license.

Constraints

CommercialUse: this constraint is related to the target of the uses of the work. It can be restricted in order to disallow that rights may be exercised for commercial purposes.

There are not more constraints defined in the language because they are not required for CC purposes. For instance, there are not user based constraint because the target of CC governed works is always the general public of Internet users. Therefore, any differentiation based on user has no sense and they are not included in that language.

Conditions

Creative Commons agreement take place outside the CC environment as it does not complement a DRM. For instance, the agreement can take place offline and it is also controlled offline by Copyright law. Therefore, CC has no payment elements at all in its rights expression language.

CC just defines some usage constraints. The constraints are quite particular to Creative Commons because they are taken from the licensing of open source materials domain. Both fields have the same objective, i.e. to promote sharing within the community. Therefore, to share is a requirement of use, together with other conditions that encourage moral rewards for authors and to facilitate re-use by providing the source from which the work was produced. These are the concrete usage conditions:

- ***Attribution***: credit must be given to copyright holder and/or author.
- ***Notice***: copyright and license notices must be kept intact.
- ***ShareAlike***: derivative works must be licensed under the same terms as the original work.
- ***SourceCode***: source code (the preferred form for making modifications) must be provided for all derivative works.

5.4 Open Digital Rights Language (ODRL)

ODRL[2] was developed trying to build an open standard for expressing machine-readable licenses for digital materials, i.e. a rights expression language. Although ODRL was initially a development of the IPR Systems enterprise, it is now an open and cooperative project with many participating

[2] http://www.odrl.net

organisation. ODRL consists of an expression language and a data dictionary. Each of the ODRL parts is defined by an XML Schema, so ODRL is based on XML metadata, see the XML section 3.4.1.

The expression language part, called ODRL-EX, defines the basic terms to be included in rights expressions and how they are organised, i.e. the syntax of the language. The data dictionary part, called ODRL-DD, is in charge of defining more concrete terms for the language, which are build on top of the general ones provided by ODRL-EX. The ODRL-DD is also an example of how ODRL can be extended using XML Schema mechanisms. These extension can be performed starting from the basic expression language terms, but also from the data dictionary terms.

ODRL is intended to be machine-actionable as part of a digital rights enforcement system. Consequently, as it is shown in the next subsections, ODRL is a control-oriented REL. ODRL has an open license, so it is provided for free use by anyone who wishes to incorporate all or part of it into their own DRM system. A reduced profile of ODRL has been adopted by the Open Mobile Alliance (OMA) in order to provide a standard digital rights management language for the mobile communications domain. OMA is formed by nearly 200 companies including the world's leading mobile operators, device and network suppliers, information technology companies and content and service providers, e.g. Ericsson, Nokia, IBM, Microsoft, Alcatel, NTT DoCoMo, etc.

5.4.1 Contract

The intention is to make ODRL a fully machine-readable language that supports digital rights enforcement and end-to-end supply chain services. However, although the ODRL REL can support a machine-actionable rights management system, it does not provide such a system nor does it make any statement about how that system should work. Therefore, it can be said that ODRL is a general REL that is independent from the concrete implementation, i.e. DRM System, that will employ it.

Contracts are modelled using three different main terms:

- ***Offer***: ODRL supports expressing offers made from Rights Holders for specific rights over their assets. The *Offer* entity allows for detailed expressions of particular Rights Holders who have negotiated and agreed to offer particular permissions over their assets.
- ***Agreement***: ODRL supports expressing agreements made between parties for specific rights over assets. The *Agreement* entity allows for detailed expressions of particular parties who have negotiated and agreed to a set of particular permissions over some assets.
- ***Revoke***: ODRL supports revoking offers, agreements, and other rights expressions. The *Revoke* entity allows for the specification, via a unique identifier, of the rights expression that is

being revoked. Unique identifier can be assigned, using the *Context* element, to entire rights expressions, offers, agreements and permission. Therefore, any or all of them can be revoked, even in the same revoke statement.

Finally, contracts can be also digitally signed. In this case, the necessary metadata for signature validation is attached to the contract using the *Signature* element, which specifies the signed info, signature value and signature key info.

5.4.2 Control

ODRL relies on external means to implement access controls, so they are outside of the scope of this language. The focus is on usage controls, although usage controls are only realized when ODRL is used in a secure and trusted systems environment. Thus, ODRL is suitable to be used for usage control but its adoption does not imply that such controls are in place. It needs a DRM System that implements these controls.

In order to implement control, ODRL relies on unambiguous definition of the available rights, which must be connected to the different actions governed by the implementing DRMS. Another important point are constraints, which should be properly quantified in order to be interpreted by the DRMS. For instance, it is important to define resource units (pages, chapters, etc.) or expression of time units and intervals. There are more details about the means provided by ODRL in order to implement control in the next section, where the language elements are described.

5.4.3 Data Elements

The data elements defined by ODRL are specified in the following subsections depending on their category as specified in the RELs Analysis section 5.2.3. There is an example of ODRL license in Table 5.2. It is important to highlight the *context* element because it is widely used to attach identifiers and descriptive metadata to any of the resources taking part in a license, e.g. licenses locations, parties names, etc. It is also possible to use metadata from external schemas, e.g. parties roles from MARC (http://www.loc.gov/marc), resources resolution from MPEG-7 [101], etc.

Agents

ODRL uses the *party* element in order to specify the agents participating in a license, both end-users and rights holders. There are also some general elements that can be employed to give more details

```xml
<?xml version="1.0" encoding="UTF-8"?>
<o-ex:rights xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
xmlns:o-dd="http://odrl.net/1.1/ODRL-DD">
<o-ex:agreement>
<o-ex:context>
<o-dd:uid>urn:ebook.world/999999/license/1234567890-ABCDEF</o-dd:uid>
<o-dd:pLocation>Sydney, Australia</o-dd:pLocation>
</o-ex:context>
<o-ex:asset>
<o-ex:context>
<o-dd:uid>urn:ebook.world/999999/ebook/rossi-000001</o-dd:uid>
</o-ex:context>
</o-ex:asset>
<o-ex:permission>
<o-dd:display>
<o-ex:constraint>
<o-dd:cpu>
<o-ex:context>
<o-dd:uid>Adobe-WebBuy:CPD-ID:ER-393939-DSS-787878</o-dd:uid>
</o-ex:context>
</o-dd:cpu>
</o-ex:constraint>
</o-dd:display>
<o-dd:print>
<o-ex:constraint>
<o-dd:count>2</o-dd:count>
</o-ex:constraint>
</o-dd:print>
<o-ex:requirement>
<o-dd:prepay>
<o-dd:payment>
<o-dd:amount o-dd:currency="AUD">20.00</o-dd:amount>
<o-dd:taxpercent o-dd:code="GST">10.00</o-dd:taxpercent>
</o-dd:payment>
</o-dd:prepay>
</o-ex:requirement>
</o-ex:permission>
<o-ex:party>
<o-ex:context>
<o-dd:uid>urn:ebook.world/999999/users/msmth-000111</o-dd:uid>
<o-dd:name>Mary Smith</o-dd:name>
</o-ex:context>
</o-ex:party>
</o-ex:agreement>
</o-ex:rights>
```


Table 5.2: Open Digital Rights Language license example

about parties, independently of their kind, e.g. the party identifier or its name. However, in some cases it is important to provide metadata specific to the kind of party. For instance, Rights Holders covers roles such as creators, producers, distributors, etc., which can be specified using the *context* element and external metadata elements, e.g. from MARC or Dublin Core.

There is also one element in ODRL that is specific to parties of the rights holders kind. It is called *rightsholder* and it is used to specify royalty entitlements, which are specified using the following elements:

- ***percentage***: indicates a payment due to the indicated party for each transaction over the asset as a percentage of the value of the net transaction.
- ***fixedAmount***: indicates a payment due to the indicated party for each transaction over the asset as a fixed value of the net transaction.

Resources

In order to specify the resources that take part in ODRL licenses, there is the *asset* element. It is complement, as for agents, with the *context* element in order to specify the resource identifier and descriptive metadata, e.g. name, resolution, etc. It is also possible to include cryptographic metadata for protected resources using the *digest* and *KeyInfo*.

Rights

The rights that are granted by a ODRL license are related to it using the *permission* element. This element specifies the concrete rights that are authorised, which are detailed in the next subsections following the classification established in the RELs Analysis section 5.2.3. Moreover, it is possible to use the *Exclusivity* attribute in order to state that the included rights are granted exclusively and the *context* element to provide and identifier to the concrete rights package inside the *permission* element. These identifiers can used to revoke a concrete rights package.

Manage This category indicates a set of digital asset management operations. The actions for the digital management of an asset are:

- ***delete***: the act of deleting a copy of an asset.
- ***install***: the act of allowing for the operation of loading, verification and certification of an asset into a data storage device.

- ***move***: the act of allowing a digital asset to move between data storage devices. Specification of constraints on the data storage devices may be allowed.
- ***uninstall***: the act of allowing for the removal from or disabling of an asset in a data storage device.
- ***duplicate***: the act of making an exact copy of a digital asset between data storage devices. Specification of constraints on the data storage devices may be allowed.
- ***backup***: the act of making copies of an asset for the purpose of guarding against the loss of the original due to accident or catastrophic media or equipment failure.
- ***verify***: the act of allowing authorization to check the authenticity of an asset.
- ***restore***: the act of allowing the conversion of a backup copy into a usable copy in a controlled manner.
- ***save***: the act of saving a copy (including any changes) of an asset to permanent storage.

Re-use It indicates a set of operations in which the asset, or portions of it, can be re-utilised. The actions for the re-utilisation of an asset creating a new asset are:

- ***modify***: the act of changing parts of the asset creating a new asset.
- ***excerpt***: the act of extracting (replicating) unchanged parts (or all) of the asset for reuse into another asset.
- ***annotate***: the act of adding notations/commentaries to the asset creating a new asset.
- ***aggregate***: the act of using an asset (or parts of it) as part of a composite work or collection.

Transfer This category indicates a set of procedures in which the rights over the asset can be transferred. The actions for the downstream transfer of rights of an asset are:

- ***sell***: the act of allowing the asset to be sold (ownership transfer) in exchange of value.
- ***lend***: the act of allowing the asset to be made available for temporary use then returned (without exchange of value). During this period, the asset is only available to the lendee. Temporal constraints are required for downstream use.
- ***give***: the act of allowing the asset to be given away (ownership transfer) in perpetuity without exchange of value.

- ***lease***: the act of allowing the asset to be made available for a fixed period of time then returned (for exchange of value). During this period, the asset is only available to the lessee. Temporal constraints are required for downstream use.

Use It indicates a set of methods in which the asset can be consumed. The actions pertaining to the end-use of an asset are:

- ***display***: the act of rendering the asset onto a visual device.
- ***execute***: the act of executing the asset. For example, machine executable code or Java.
- ***play***: the act of rendering the asset into audio/video form.
- ***print***: the act of rendering the asset onto paper or hard copy form.

Constraints

ODRL supports the expression of rights constraints, which restrict the permitted actions over the license asset. In ODRL, a constraint is associated with one permission. If a constraint appears at the same level as a number of permissions, then the constraint applies to all of the permissions. Constraints can also have constraints. In this case, the child constraint applies to the parent constraint. As an example of this, consider the *unit* constraint and it's meanings when used within the *count* and *range* constraints:

- A *count* constraint containing a *unit* constraint means the number of times the right can be exercised, e.g. a *count* of 5 inside a *print* element means "print 5 times".
- A *unit* constraint containing a *count* constraint means the number of units, e.g. a *count* of 5 inside a *NumberOfPages* type *unit* element that is itself inside a *print* element means "print 5 pages".
- A *unit* constraint containing a *range* constraint means the minimum or maximum ordinal position of the units, e.g. a *range* with *min* of 1, *max* of 100, inside a *NumberOfPages* type *unit* element that is itself inside a *print* element means "print pages numbered between 1 and 100".

Additionally, all constraint elements may have a *context* element to support the use of identifiers and a *type* attribute to refer to additional information. Finally, it is important to note that any constraint that is expressed but cannot be performed by the implementing DRM system, must not be granted. That is, if a system does not understand how to guarantee that a specified constraint be honoured it must not grant the permission at all. For permissions with multiple constraints, all constraints must be honoured with no conflicts arising.

The constraints are grouped in the following categories.

User This category indicates a set of constraints which limits usage to identified users, which are specified individually or as a group:

- ***individual***: an identifiable party acting as an individual. *Context* is used to identify the individual.
- ***group***: a number of identifiable parties acting as a collection of individuals. *Context* is also used to provide a group identifier.

Device It indicates a set of constraints which limits usage to physical devices or systems. Device constraints apply to any electronic or digital equipment or system:

- ***cpu***: specify an identifiable computing system with a central processing unit (CPU) using *context* to provide the identifier.
- ***network***: an identifiable data network. *Context* is used to identify the device and *range* to indicate the IP address restriction.
- ***screen***: an identifiable display output screen device. For example, a screen reader or braille device. *Context* is used to identify the device.
- ***storage***: specify an identifiable storage media device. For example, a hard disk or removable cartridge. *Context* is used to identify the device.
- ***memory***: an identifiable memory device. For example, the clipboard. Memory devices are identified using the *context* element.
- ***printer***: specify an identifiable hard copy printer using the *context* element.
- ***software***: an identifiable software application that must be present in order to enable the restricted action. The application is referenced by an identifier in a *context* element.
- ***hardware***: specify an identifiable generic hardware device using the *context* element.

Bounds This category indicates a set of constraints which limits usage to a fixed number or extent/coverage. These are the bound constraints that define limits/extent within which any entity can function:

- ***count***: a numeric count indicating the number of times the corresponding entity may be exercised. The value is a positive integer, for example, the *print* usage may be constraint with a *count* of 10 meaning that the *asset* can be printed zero to 10 times.
- ***range***: a numeric range indicating the min/max values of the corresponding entity that the constraint applies to. Contains the *min* and *max* subelements, which respectively define the beginning of the range and the end of the range, both inclusive. The numeric values must use the ordinal position when referring to external objects. Positive and negative decimals must be supported. If there is no *min* or *max* value, then the range is open-ended. For example, a *min* of "1" and no *max* means that the *range* has an unlimited maximum.
- ***spatial***: used to restrict the geographic area where the restricted action is enabled. *Context* is used to identify the spatial area with codes specified from a controlled vocabulary, e.g. ISO3166 for country codes.

Temporal It indicates a set of constraints which limits usage to temporal boundaries. These are the temporal constraints for the time limits within which any entity can function:

- ***datetime***: a date and/or time-based range. It can contain the *start*, *end* and *fixed*. They respectively define the the beginning of the range, the end of the range and an exact point in date/time. For ranges the values are inclusive and date and time values must conform to ISO8601. If there is no start and/or end value, then the range is open-ended.
- ***accumulated***: the maximum period of metered usage time. Period value must conform to ISO8601. For example "P30H" indicates a 30 hour period.
- ***interval***: recurring period of time in which the rights can be exercised. Date and Time value must conform to ISO8601. For example "P7D" indicates a 7 day period.

Aspect This category indicates a set of constraints which limits usage to distinct features or expressions of the asset. These are the aspect constraints for distinct features of the asset:

- ***quality***: specification of constraints on the quality aspects of the asset. Contains the *type* attribute for the classification of the quality type. The values for the type attribute must be from a well known vocabulary and represented as a URI. For example, the resolution of an image or number of colours.

- ***format***: specification of constraints on the formats of the asset. Contains the *type* attribute for the classification of the format type. The values for the type attribute must be from a well known vocabulary and represented as a URI. For example, values can taken from the Internet Media Type list[3].
- ***unit***: specification of constraints on the whole asset or sub-parts of the asset. Contains the *type* attribute for the classification of the sub-unit part type. the values for the type attribute must be from a well known vocabulary and represented as a URI.
- ***watermark***: specification of watermarking requirements for the asset. Use *context* to identify the watermark information.

Target It indicates a set of constraints which limits usage to where and how the asset is used. These are the target constraints to specify how and where limits over the asset:

- ***purpose***: specification of a specific purpose to which the usage is constrained. Use *context* to identify the purpose from a known vocabulary.
- ***industry***: specification of a specific industry group to which the usage is constrained. Use *context* to identify the industry from a known vocabulary.
- ***recontext***: specification if the asset may or may not be re-contextualised using a boolean value.

Rights This category indicates a set of constraints which only applies to assets with a transfer permissions and enables the specification and constraints on downstream permissions.

- ***transferPerm***: specification of constraints over assets for which there is a downstream transfer of permissions. It applies only to assets that have one of the transfer permissions applied, i.e. *sell*, *lend*, *give* or *lease*, and it limits the extent to which the authorised party can transfer the received rights to other users. Contains the *downstream* attribute that defines the allowable narrowing of the specified permissions. The values are *equal*, *less* or *notgreater*. They mean, respectively, that the permissions must be passed along without change when the asset is transferred, that a smaller subset of permissions must be passed and that the permissions may be narrowed but they must not be expanded.

[3]http://www.iana.org/assignments/media-types

Conditions

ODRL defines a set of conditions that must be fulfilled in order to exercise the granted rights. They are called requirements in the ODRL nomenclature. There is a set of payment types that could be converted to machine-enforceable requirements:

- ***payment***: the amount of the payment. It contains the *amount*, *currency*, *taxpercent* and *code* subelements. The *amount* must be a positive decimal to two decimal places. The mandatory *currency* must use ISO4217 codes. The *taxpercent* must be a positive decimal between 0 and 100 inclusive, with an optional tax code identifier.
- ***prepay***: the amount due prior to the granting/use of the rights. The payment element is used to specify the amount together with optional temporal constraints.
- ***postpay***: the amount due after the use of the rights. The payment element is used to specify the amount together with optional temporal constraints.
- ***peruse***: the amount due for each use of the granted rights. The payment element is used to specify the amount.

Aditionally, ODRL specifies some interaction requirements, i.e. obligations in the form of user actions:

- ***accept***: user must view and agree to textual information.
- ***register***: user must register their details with a service provider.

And there are also usage conditions:

- ***attribution***: the use of the asset must always include attribution of the asset owners.
- ***tracked***: the user will be tracked for their use of the asset. The user must be aware of privacy policy of the service provider.

5.5 MPEG-21

MPEG-21 [22] is a suite of standards relating to digital multimedia resources. There are sixteen parts to MPEG-21, including identification of digital items, content representation, delivery protocols and content management. The latter is comprised of a rights expression language and a a data dictionary, parts 5 and 6 respectively.

Part 5 of is the Rights Expression Language [120]. It defines the basic terms to be included in rights expressions and how they are organised, i.e. the syntax of the language. The REL is specified by three main XML Schemas:

- **Core**: this XML Schema specifies the basic elements of the language, e.g. *Right*, *Resource*, *LicensePart*, etc.
- **Standard Extensions**: it defines the basic extensions to the core part, e.g. *Territory*, *EmailName*, *Rate*, etc. The XML Schema extension mechanism are used to build these definitions on top o those from the core.
- **Multimedia Extensions**: this is the more specific XML Schema because it is build on top of the previous ones. It defines the multimedia specific terms of the language, e.g. *Play*, *Renderer*, *Mark*, etc.

Part 6 is the Rights Data Dictionary (RDD) [120]. The objective of this part is to define the terms used in the REL expressions, i.e. to formalise the semantics of the language terms. Differently to the ODRL data dictionary, MPEG-21 RDD is not specified using XML Schemas. RDD is an ontology, see the Ontology section 4.8.5, although it uses a semiformal language for its specification. The terms used in the REL, plus additional terms, are defined by semantic interrelationships. This ontology can be used to facilitate the implementation of MPEG-21 tools for content management. However, MPEG-21 REL and RDD are not directly integrated, as they are specified using different approaches, and some effort is needed in order to implement the RDD because it is not formalised.

In this analysis, in order to facilitate comparing the different analysed RELs, we are going to centre the MPEG-21 study on the REL. The MPEG-21 REL is oriented towards the licensing of digital materials. As it has been presented in the RELs History section 5.2.1, it was developed by MPEG-21 standards group using XrML as its basis. The standard is specifically intended to be unambiguously machine-actionable and to interact with software and hardware that will enforce the license permissions, i.e. a DRM System. The latter will provide the necessary implementation using trusted systems technology which will allow end-to-end control over digital works through the whole content value chain.

The REL standard is broad enough to make it usable for a wide variety of digital products. It provides mechanisms for its extension and adaptation to concrete application domains. For instance, the Open eBook Forum[4], an industry group developing standards for e-books, is considering a set of extensions to MPEG21 REL specific to e-books.

[4]http://www.openebook.org

5.5.1 Contract

MPEG-21 REL is quite similar to ODRL. It is also a control-oriented language so it describes a machine-to-machine language for automated license control and management. The final objective is to provide a key component for a platform that supports the secure delivery of content over digital networks. Within this environment, the rights expression language is only one of sixteen architectural elements. And, also like ODRL, this REL does not describe or define the system that will make use of it. The language will be used within the context of a trusted system that can operate along the entire e-commerce business chain to manage business relationships as well as end-user permissions.

MPEG-21 REL contracts relate one or more principals, a set of rights that are associated with a digital resource, and conditions to which those rights are subject. A principal can be a person, a network node or an end-user device. A right is described in its linguistic role as a "verb". The resource is the object of the rights and a condition describes rules under which rights can be exercised. Moreover, the object of the contract, i.e. the resource, can be a rights expression. Therefore, the REL can be used to model the transfer of rights along the full chain of electronic commerce.

For instance, it can be used to carry to the digital world the flow of contracts from rights holders to other parties like wholesalers and retailers interact in the analogue world.

5.5.2 Control

The MPEG-21 REL rights language is designed specifically for systems that will be imbedded in devices or software and that will exercise control over the uses of the digital file. MPEG-21 REL focuses on usage controls, and access controls are outside of the scope of this language. To achieve flexibility in the control of the different kinds of usages, MPEG-21 has abstracted the kinds of actions that can be performed on creations, which are called rights, in the Rights Data Dictionary (RDD). The rights terms are taken from the controlled list of verbs in RDD. All of them derive from the generic verb "do".

For instance, the rights terminology in the area of allowable changes to a file is based on the verb "modify" and subdivided into "enlarge", "reduce" and "move". Another example of abstraction is the term "play" to represent all of the possible ways that a resource can be transiently rendered for human perception, while "print" refers to a fixed perceivable rendering.

Although this refinement is possible, to date, MPEG-21 REL uses only a small set of verbs and does not take profit from their hierarchical organisation in the dictionary. As it has been previously commented, there is not a standard mechanism for REL to RDD interoperation. Therefore, the common approach for the moment is to implement just the basic verbs that are present in the REL. The hierarchical organisation does not contribute any additional meaning as the REL terms are quite distant in the complete hierarchy provided by the RDD and not hierarchically related among them.

5.5.3 Data Elements

The data elements defined by MPEG-21 REL are specified in the following subsections depending on their category as specified in the RELs Analysis section 5.2.3. There is an example of MPEG-21 REL license in Table 5.3. The main element is *license* that encapsulates the whole license. It is composed of two sections. The first one, marked by the *grant* element, is the main one and it contains the actual content of the license, i.e. the granted right, the agents involved, the governed resource and the conditions. The other section is marked by the *issuer* element and it is in charge of stating who is the issuer of the license. It is not necessary that the isssuer party is involved in the grant part of the license.

Agents

MPEG-21 REL refers to the agents involved in a license using the general term *principal*, which can refer to any person, entity, or system component. Principals appear in the *issuer* part of the license, as license issuers, or in the *grant* part, as users or rights holders affected by the rights statement. Principals are identified by a name or an encrypted key. The latter identifies but also authenticates the principal and is specified by the *keyHolder* element, which details the encrypted key values, e.g. for a *RSAKeyValue* there are the *Modulus* and *Exponent* subelements.

Resources

Resources in MPEG-21 REL are defined as digital items, which constitute another part of the MPEG-21 standard. The link between the REL expressions and the digital items is done using the *diReference* element. This element specifies the concrete resources through a *identifier* subelement. There are not additional means in the REL for resource description. These are supposed to lay outside the scope for MPEG-21 REL.

For instance, as in the case of ODRL, MPEG-7 metadata can be used to describe digital multimedia resources details.

```
<?xml version="1.0" encoding="UTF-8"?>
<r:license xmlns:r="urn:mpeg:mpeg21:2003:01-REL-R-NS"
xmlns:sx="urn:mpeg:mpeg21:2003:01-REL-SX-NS"
xmlns:mx="urn:mpeg:mpeg21:2003:01-REL-MX-NS"
xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:mpeg:mpeg21:2003:01-REL-MX-NS rel-mx.xsd">
<r:grant>
<r:keyHolder licensePartId="John">
<r:info>
<dsig:KeyValue>
<dsig:RSAKeyValue>
<dsig:Modulus>KtdToQQyzA==</dsig:Modulus>
<dsig:Exponent>AQABAA==</dsig:Exponent>
</dsig:RSAKeyValue>
</dsig:KeyValue>
</r:info>
</r:keyHolder>
<mx:play/>
<mx:diReference>
<mx:identifier>urn:grid:a1-abcde-1234567890-f</mx:identifier>
</mx:diReference>
<r:validityInterval>
<r:notBefore>2003-01-01T00:00:00</r:notBefore>
<r:notAfter>2004-01-01T00:00:00</r:notAfter>
</r:validityInterval>
</r:grant>
<r:issuer>
<r:keyHolder licensePartId="Xin">
<r:info>
<dsig:KeyValue>
<dsig:RSAKeyValue>
<dsig:Modulus>X0j9q99yzA==</dsig:Modulus>
<dsig:Exponent>AQABAA==</dsig:Exponent>
</dsig:RSAKeyValue>
</dsig:KeyValue>
</r:info>
</r:keyHolder>
</r:issuer>
</r:license>
```

Table 5.3: MPEG-21 Rights Expression Language license example

Rights

The rights that are granted by a MPEG-21 license are directly included in the *grant* section of the license. The concrete right elements for the available rights are detailed in the next subsections following the classification established in the RELs Analysis section 5.2.3.

Manage

- ***delete***: the act of destroying a digital resource that is not capable of reversal. After a delete process, an undelete action is impossible.
- ***install***: the act of following the instructions provided by an installing resource.
- ***move***: the act of relocating a resource from one place to another. With an *move*, at least the location of the resource is changed.
- ***uninstall***: the act of following the instructions provided by an uninstalling resource.

Re-use

- ***adapt***: the act of changing transiently an existing resource to derive a new resource. With an *adapt*, two distinct resources will exist as a result of the process, one of which is the original resource in unchanged form, and one of which is newly made. Changes can include the addition to and removal of elements of the original resource, including the embedding of other resources.
  This right is also used to represent the right to copy, which in this context means to derive a new resource which has the same set of specified or implied attributes as its source, as a copy with absolutely identical attributes to the original cannot logically exist.
- ***diminish***: the act of deriving a new resource which is smaller than its source. With *diminish*, two distinct resources will exist at the end of the process, one of which is the original resource in unchanged form, and one of which is newly made, whose content is adapted from the original resource, and a measure of which is smaller than that of the original. Changes can include the removal of elements of the original resource.
- ***embed***: the act of putting a resource into another resource. The resource into which a resource is embedded can be pre-existing or can be created by the act of combining the embedded resource with one or more others. Embed describes a process by which something, an embedded resource, becomes a part of something else which already exists, a host.

- ***enhance***: the act of modifying a resource by adding to it. With an *enlarge*, a single resource is preserved at the end of the process. Changes can include the addition of new material, including the embedding of other resources, but not the changing or removal of existing elements of the original resource.

- ***enlarge***: the act of deriving a new resource which is larger than its source. Similar to *diminish* but there is a measure of the new resource is greater than that of the original.

- ***modify***: the act of changing a resource, preserving the alterations made. With an *modify*, a single resource is preserved at the end of the process. Changes can include the addition to and removal of elements of the original resource, including the embedding of other resources.

- ***reduce***: the act of modifying a resource by taking away from it. With an *reduce*, a single resource is preserved at the end of the process. Changes can include only the removal of existing elements of the original resource.

Transfer

- ***tansferControl***: the generic act of transferring the rights over a resource. It can be concretised using constraints in order to model typical transfer rights. For instance, it can be combined with a time-limited constraint and a new attribute that signals that the transaction is of the "lend" type in order to model a right similar to the lend one present in ODRL.

Use

- ***execute***: the act of executing a digital resource. An *execute* refers to the primitive computing process of executing.

- ***play***: the act of deriving a transient and directly perceivable representation of a resource. A *play* covers the making of any forms of transient representation that can be perceived directly (that is, without any intermediary process) with at least one of the five human senses. It includes playing a video or audio clip, displaying an image or text document, or creating transient representations that can be touched, or perceived to be touched. When an *play* is applied to a digital resource, content can be rendered in any order or sequence according to the technical constraints of the digital resource and renderer.

- ***print***: the act of deriving a fixed and directly perceivable representation of a resource. An *print* refers to the making of a fixed physical representation, such as a hard-copy print of an image or text, that can be perceived directly (that is, without any intermediary process) with one or more of the five human senses.

Constraints

MPEG-21 uses the same term, condition, for both the constraints and the conditions as they are understood in this analysis. Here, constraints are interpreted as conditions that restrict the rights granted by the corresponding license. However, in MPEG-21 REL, constraints and conditions are grouped under the MPEG-21 REL "condition" category. In order to facilitate comparing the different RELs, the MPEG-21 REL conditions that are related to rights restrictions have been analysed in this section. For the rest of the MPEG-21 REL conditions, which are interpreted as "conditions" from the point of view of this analysis, are presented in the corresponding section.

The constraints are grouped in the following categories:

Device This category groups the constraints which limit usage to physical devices or systems. MPEG-21 REL does not directly define any device constraint. However, the e-Book Forum extension to MPEG-21 REL defines two device constraints:

- ***clipboard***: when this constraint is specified, the device clipboard service is disabled so it is not possible to copy the governed resource completely or in part to the clipboard.
- ***textToSpeechOff***: when this constraint is specified, the text to speech functionality must be disabled in the user device.

Bounds This category indicates a set of constraints which limits usage to a fixed number or extent/coverage. These are the bound constraints that define limits/extent within which any entity can function:

- ***exerciseLimit***: the right can be exercised a number of times determined by the following conditions: if the *count* subelement is present, the invocation of the service specified by *serviceDescription* returns a value that is less than or equal to the value *count*, or, if *count* is absent, the invocation of the service returns true.
- ***trackQuery***: the right is constrained to those situations where the value returned by the invocation to the service referenced by *serviceDescription* is both greater than or equal to the value of *notLessThan*, if *notLessThan* is present, and less than or equal to the value of *notMoreThan*, if *notMoreThan* is present.

- ***territory***: this constraint limits the right to locations specified by the tuple (C,r,S,c,p,s) for which all of the following conditions on the constraint subelements are true: if *country* is present, its value is C; if *region* is present, its value is r; if *state* is present, its value is S; if *city* is present, its value is c; if *postalCode* is present, its value is p; and if *street* is present, its value is s.

Temporal It indicates a set of constraints which limits usage to temporal boundaries. These are the temporal constraints for the time limits within which the grant is valid:

- ***validityInterval***: if the *notBefore* subelement is present, the start time of the governed action must be greater than or equal to the instant in time represented by the value of *notBefore*, and if *notAfter* is present, the end time of the governed action must be less than or equal to the instant in time represented by the value of *notAfter*.
- ***validityIntervalFloating***: this constraint specifies a *duration* within which the usage must be carried out, otherwise it is not valid.

Aspect This category indicates a set of constraints which limits usage to distinct features or expressions of the asset. These are the aspect constraints for distinct features of the asset:

- ***diPartOf***: two resources are referenced by *diReference* and *identifier*, the constraint is satisfied if the first one is part of the second one.
- ***isMarked***: the right can only be exercised if the resource referenced by *diReference* and *identifier* is marked with the mark specified by *watermark*.

Conditions

MPEG-21 REL has the most detailed language for payments of all of the studied languages:

- ***FeeFlat***: it requires the satisfaction of an amount of money that is greater or equal in amount to and of the same currency as the amount of money indicated by *rate*.
- ***FeePerUse***: it requires that a fixed amount of money is satisfied for each use. The amount of money satisfied must bes greater or equal in amount to and of the same currency as the amount of money indicated by *rateamount* and *ratecurrency*.

- ***FeeMetered***: it requires that an amount of money dependant on the consumption time is satisfied. The required amount must be equal or greater to and of the same currency as: x * (y/z) * (floor(w/y) + r). Where x is the amount of money indicated by the *rate* subelement, y is the numerical value of the subelement *by* in seconds, z is the numerical value of the subelement *per* in seconds, w is the numerical value of the duration of the governed act in seconds and r is "1" if w modulus y is greater than the numerical value of the subelement *phase* in seconds and 0 otherwise.
- ***FeePerInterval***: it requires that for each interval of time specified by the *per* element the required amount of money is satisfied. The amount is specified by the *amount* and *currency* subelements of the *rate* element.
- ***FePerUsePrePay***: it requires the same amount of money than *FeePerUse* but the money must be satisfied prior to the use.

And there are also usage conditions:

- ***SeekApproval***: this condition specifies a service description by reference. The description specifies how to invoke the service that must return a true value in order to enable the use conditioned by *SeekApproval*.

5.6 RELs Overview

CC	MPEG-21 REL	ODRL
	Agent Data Element	
Agent	principal, issuer	party
	Manage-type Rights	
	delete	delete
	install	install
	move	move
	uninstall	uninstall
		duplicate
		backup
		verify
		restore
		save

Reuse-type Rights		
DerivativeWorks	adapt	
	diminish	
	embed	
	enhance	
	enlarge	
	modify	modify
	reduce	
		excerpt
		annotate
		aggregate
Transfer-type Rights		
Distribution		
		sell
		lend
		give
		lease
	transferControl	
Use-type Rights		
Reproduction		
		display
	execute	execute
	play	Play
	print	print
User Constraints		
		individual
		group
Device Constraints		
	clipboard	
	textToSpeechOff	
		cpu
		network
		screen
		storage

		memory
		printer
		software
		hardware
Limits Constraints		
	exerciseLimit	count
	trackQuery	range
	territory	spatial
Temporal Constraints		
		datetime
		accumulated
	validityInterval	interval
	validityIntervalFloating	
Aspect Constraints		
	diPartOf	quality
		format
		unit
	isMarked	watermark
Target Constraints		
CommercialUse		purpose
		industry
		re-context
Payment Conditions		
	feeFlat	payment
	feePerUsePrePay	prepay
		postpay
	feePerUse	peruse
	feeMetered	
	feePerInterval	
Usage Conditions		
	seekApproval	
attribution		attribution
notice		

shareAlike		
		tracked

Table 5.4: Overview of the RELs data elements

5.7 Other RELs

There other rights expression languages. They are shortly described in the next subsections.

5.7.1 OntologyX

OntologyX[5] is, as its name shows, an ontology, like MPEG-21 RDD. It is the initiative of a private company, RightsCom[6], and there is not very much public information about it. OntologyX is presented as a formal ontology that has been implemented using OWL, see Ontology section 4.8.5. Therefore, it is not based on a language definition as almost all the existing RELs. The ontology builds a conceptual model that can be divided in the following submodels.

Content Model

It defines the main entities in the content and Digital Rights Management domain:

- ***Abstraction***: a song or an album.
- ***Performance***: a recording, which is based on an abstraction.
- ***Physical Fixation***: a CD, which is based on a performance and is physical.
- ***Digital Fixation***: an MP3, which is based on a performance and is digital.

Context Types

This is the main component of the OntologyX model. It defines a set of context that are related to an event based approach. Each event modelled by OntologyX, which are DRM related events, is formalised as a context where the different entities taking part in the event are grouped. These are the main contexts:

[5] http://www.ontologyx.com
[6] http://www.rightscom.com

- ***Assignment***: how rights are assigned and delegated. The entities taking part in this context are: assigner, assignee, assignedRight, assignedResource, time and place.
- ***Rights Statement***: claims to control particular rights. The participants in this context are: rightsController (= Assignment/assignee), controlledRight (= Assignment/assignedRight), controlledResource (= Assignment/assignedResource), time and place.
- ***Proposal***: offers and requests for how rights can be used. The entities taking part in this context are: offerer (= RightsStatement/rightsController), permission, prohibition, requirement, time and place.
- ***Agreement***: agreer1 (= Proposal/offerer), permission, prohibition, requirement, agreer2, time and place
- ***Permission***: allowed use event. In terms of deontic logic, see Deontic Logic section 2.6.1, it is interpreted as permission ($\Diamond$). The participants in this context are: user (= Agreement(agreer2), usedResource (= RightsStatement/controllerResource), exception (= Prohibition), condition (= Requirement), time and place.
- ***Prohibition***: prohibited use event, an exception to a permission. In terms of deontic logic it is interpreted as not permission ($\neg\Diamond$). The entities taking part in this context are: user (= Permission/user), usedResource (= Permission/usedResource), time and place.
- ***Requirement***: something required to happen, i.e. the condition of a permission. Usually, it is a payment. In terms of deontic logic it is interpreted as obligation ($\Box$). The participants in this context are: payer (= Permission/user), requiredResource, time and place.

There is a common flow of the previous events, i.e. context. The starting point is a situation of controlled rights generated by an Assignment and captured by a Rights Statement. The controlled rights define permissions and prohibitions that take place in particular events. Then, some of this events are made explicit through an offer, i.e. a Proposal. It there is an Agreement, then particular use events can take place, which are permitted. There may be some exceptions to these permissions, i.e. Prohibitions, and in order to carry out the use something might by required, i.e. a Requirement.

5.7.2 Adobe Content Manager (ACM)

Adobe Content Manager[7] is one of the few fully implemented RELs in use today. It can be used only in the Adobe Reader product for protected files. The REL has a small vocabulary but covers the basics of printing, copying, lending, and text-to-speech.

[7] http://www.adobe.com/security/doccontrol.html

5.7.3 Publishing Requirements for Industry Standard Metadata (PRISM)

PRISM[8] is being used primarily by newspaper and magazine publishers to exchange information about articles and other elements (photos, charts) that can be re-used by other publications. In addition to descriptive metadata, PRISM includes some rights metadata elements. They are few because they were created as they were necessary for their specific and immediate needs. The rights terms are encapsulated in what is called the PRISM Rights Languag (PRL). PRISM is based on RDF metadata, see RDF Model and Syntax section 4.8.3, for resource description and rights expressions.

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

Part II

Preparation

Chapter 6

Objectives

From the general point of view, the objective of the research documented in this work is to apply the potential of Knowledge Representation in the Web context. The important thing here is to detect the problematic issues detected in current web applications and to take profit from the new possibilities of Semantic Web tools. More concretely, the idea is to apply Semantic Web technologies to electronic commerce of multimedia contents and to the management of the copyright of these contents. Both application fields are deeply interrelated so they will be considered in conjunction in this work.

The Digital Rights Management and Rights Expression Languages analysis presented in the Rights Expression Languages chapter 5.2 has highlighted some of the limitations of existing DRM initiatives and their associated RELs. First, there is a potential problem with the rights expression languages patent by ContentGuard. This patent covers any usage rights grammar. Therefore, any approach that is not syntax-based can potentially overcome the limitations imposed by such a generic patent.

Moreover, language grammars are not powerful enough to model some copyright statements, e.g. fair use, and they also make implementation more difficult due to their expressive limitations. They capture just a part of the language specification while the rest is captured in non-formal form. The captured part is the language syntax while the semantics are described in human-readable form. Thus, the semantics should be interpreted from this non-formal specification each time a REL processor is implemented or maintained.

Another important point is about Digital Rights Management Systems. As it has been shown, in order to build trusted systems, they must impose strong control over the end-user device and avoid piracy. However this is a very intrusive approach that leads to unsatisfactory user experiences, even in completely "legal scenarios". For instance, there can be interoperability problems between devices and contents governed by different DRMS or actions authorised by copyright law can be blocked by a DRMS, e.g. private copy.

This is also due to the expressive limitations of the technologies employed to date. RELs do not take into account the legal framework where the DRM systems that employ them operate. The copyright framework is too complicated to be captured by syntax-based tools. Until now, RELs are based on languages that just model the set of actions that can be authorised or prohibited to be performed on a given content for a given context. This usages model must be complemented with the legal framework that effectively governs it in the last term.

Finally, as it shown in the Control section 5.2.3, RELs have a problem with the level of detail they must support in order to cope with usage control. They have to specify every possible allowable usage in their grammar, or at least provide expansion points for this. However, at last, these expansion points must be completely detailed in order to fully specify the concrete action at hand. This is due to the lack of generalisation mechanism that is inherent to the language grammars used to formalise the RELs.

Therefore, the objective of this work is to face the disadvantage of existing Digital Rights Management Systems, which are due to the limitations of syntax-based rights expression languages. In order to face this objective, the state of the art has shown that knowledge representation techniques allow facing the previous inconveniences with guarantees. They come from a long tradition, previous to the appearance of computers, but that has shown its usefulness in the computers era.

The application context is also important and it must be taken into account. The Internet, and more concretely the World Wide Web, is the place where the more exigent requirements for DRM are being placed. Consequently, it is necessary to face the previous problems with this context in mind. As it has been also shown in the Web Technologies chapter 3, the World Wide Web has some limitations. The Semantic Web is conceived as a way to overcome them employing knowledge representation techniques, as it has been detailed in the Semantic Web chapter 4. Therefore, it seems appropriate to base this work on the Semantic Web in order to make a contribution to REL and DRM that incorporates the advantages of knowledge representation and is able to put them naturally into practice in the context of the Internet and the World Wide Web.

To summarise, the objective is to build an ontology that captures a great part of the copyright domain semantics and allows modelling rights expression that overcome current RELs limitations. Moreover, the target ontology is a Semantic Web one, which would smoothly integrate in the World Wide Web environment. This copyright ontology is going to constitute the main outcome of this work.

To conclude, it is important to note that despite the disadvantages detected in current REL initiatives, there is a great amount of work that has been carried out as it has been shown in the Rights Expression Languages chapter 5.2. A syntactic approach based on XML Schemas has been employed in order to define most of these RELs. These initiatives have performed an extensive analysis of the digital rights management field and, though the resulting grammars for the RELs capture some of the semantics highlighted during the analysis, they are not intended to formalise these implicit semantics. Consequently, they remain hidden when just syntax-based tools are employed.

An additional objective of this work is to take profit from all the hidden semantics in the existing REL grammars. In order to do that, it is necessary to formalise them using convenient tools, e.g. web ontologies and Semantic Web tools. A methodology for XML semantics reuse is going to be applied to the XML Schemas that define these grammars. This way, their implicit semantics can be made explicit and reused in order to complement the copyright ontology. Moreover, they can be also used to validate the copyright ontology.

6.1 Hypothesis

The more appropriate context for this work is the Semantic Web, because it offers an open, heterogeneous and potentially global framework for knowledge management. By automating knowledge management at the web scale, humans can obtain a greater benefit from the knowledge they globally possess. Machines can help them manage this enormous amount of information if they have access to a significant part of its intended meaning.

The working hypothesis is that knowledge representation techniques in the Semantic Web, which allow taking a semantics-based approach, will provide new opportunities for rights expressions representation and digital rights management. First of all, with these knowledge representation tools, it is possible to avoid basing DRM solutions on rights language grammars that that are very likely to be subject to patent limitations.

Moreover, these semantics-based tools have a greater expressive power. Therefore, it is possible to capture the semantics that otherwise remain implicit, formalise them and make them easily available from the computer point of view. The first consequence of this is that it will facilitate enormously DRM tools development. Additionally, it will make possible to take profit from the expressive power in order to build generalisation structures that facilitate dealing with rights situations considering just the level of detail they require.

Moreover, if we go one step further the taxonomical tools that make generalisation structures possible, there are ontologies, detailed in the Ontology section 4.8.5. Ontologies are so expressive that they will be able to capture a great amount of the underlying legal framework and combine it with the usage models typical in RELs initiatives. Therefore, it will be possible to develop a copyright ontology that takes into account copyright law together with the common usage patterns of copyrighted content.

Ontologies have the additional benefit of facilitating evolvability and interoperability. Therefore, a copyright ontology can be defined with the required level of detail for a given application context and evolve later in order to cope with new situations and requirements. And these new requirements are going to appear for sure due to the dynamism of digital technologies and global networks and markets.

On the other hand, it is quite unlikely that there is going to be just a one-fits-all solution for rights expressions representation. Therefore, interoperability is going to be a key issue and ontologies an opportunity. As ontologies do not constraint the way things are written down, i.e. the grammar, but just what are we talking about, i.e. the semantics, it is easier to interoperate. A copyright ontology will thus also facilitate interoperability among different RELs. Moreover, it will be easily enriched with the semantics that will be reused from existing initiatives, which will facilitate the development of the copyright ontology, its validation and enable it as a key tool for DRM interoperability and integration.

Chapter 7

Methodology

7.1 Overview

First of all, the premise that has guided this research work is to combine knowledge representation and web technologies in order to face digital rights management in a novel way. However, the intention is to take an approach that can be easily generalised and then applied to applications development in the Web. The first step of this approach is to analyse the application domain and construct a web ontology that captures the domain knowledge. This will be done to construct a knowledge model of the legal, commercial and management aspects of intellectual property creations and intellectual property rights. This knowledge model in the form of a web ontology constitutes the main contribution of this work. Therefore, the importance of this part justifies establishing a methodology that ensures, as much as possible, the productivity of the effort and the quality of the result.

The IEEE[1] defines methodology as "a comprehensive, integrated series of techniques or methods creating a general systems theory of how a class of thought-intensive work ought to be performed".Actually, there is not a mature knowledge engineering methodology for ontology development [32, 51]. Some of them have been developed independently but they are tailored to particular ontology developments. It lacks a mature methodology that can be widely applied to guide future ontology developments. However, one of the existing methodologies can be highlighted because it is particularly interesting for ontology development in the Semantic Web. It is Methontology, a methodology for ontology development, which is detailed in the Methontology section

[1] http://www.ieee.org

7.2. Methontology has guided the development of the copyright ontology that constitutes the fundamental contribution of this work. It is important to note that, in order to better integrate the Methontology results in the general discourse of this work, it has been used in a more narrative and less knowledge engineering oriented way.

Moreover, there are some parts of the contribution that complement the ontology. These parts extract the ontology full potential from a practical point of view. They will require software development methodologies and tools. For them, the Rational Unified Process [75], the Unified Modelling Language [14] and its evolution for agent-oriented development Agent UML [62], are the chosen alternatives. However, these methodologies and tools should be adapted to fit the particularities of the knowledge-oriented approach. Particularly, the special features of the knowledge representation technologies used to exploit the knowledge layer.

7.2 Methontology

Methontology [33] is a methodology for ontology construction. To improve its applicability it adopted some ideas from the more mature Software Engineering discipline. More concretely, its ontology development process is based on the activities identified in the IEEE standard for software development [102]. These activities are scheduled by the Methontology ontology life cycle that establishes the stages through which the ontology moves during its life time and the activities to be performed in each stage. Both parts of Methontology are detailed next.

7.2.1 Ontology Development Process

The process refers to which activities are performed when building ontologies. It identifies three categories of activities, as shown in Figure 7.1 and detailed below:

- **Ontology management activities**: they include scheduling, control and quality assurance. The *scheduling* activity identifies the tasks to be performed, their arrangement, and the time and resources needed. The *control* activity guarantees completion of tasks as intended. The *quality assurance* activity checks the quality of each methodology output (ontology, software and documentation).
- **Ontology development-oriented activities**: grouped in pre-development, development and post-development activities.
  During pre-development, the *environment* where the ontology will be used are studied and

there is a *feasibility study*.

During development, the *specification* activity states why the ontology is being built, the intended uses and the end-users. The *conceptualisation* activity structures the domain knowledge as meaningful models at the knowledge level. The *formalisation* one transforms the conceptual model into a formal or semi-computable model. Finally, the computable models are built in *implementation*.

During post-development, the *maintenance* activity updates and corrects the ontology if needed and it can be reused by other ontologies or applications.

- **Ontology support activities**: they are performed at the same time as the development-oriented activities. During support the following activities take place. The *knowledge acquisition* activity whose objective is to acquire knowledge from experts or by (semi)automatic ontology learning. *Evaluation* activity that judges the developed ontologies, software and documentation against a frame of reference. *Integration* activity if other ontologies are reused possibly in conjunction with *merging* or *alignment* activities if multiple ontologies are reused and need to be combined. Merging produces a new ontology from the combination while alignment establishes mappings that preserve the original ontologies. *Documentation* details each completed stage and product and *configuration management* records ontologies, software and documentation versions in order to control changes.

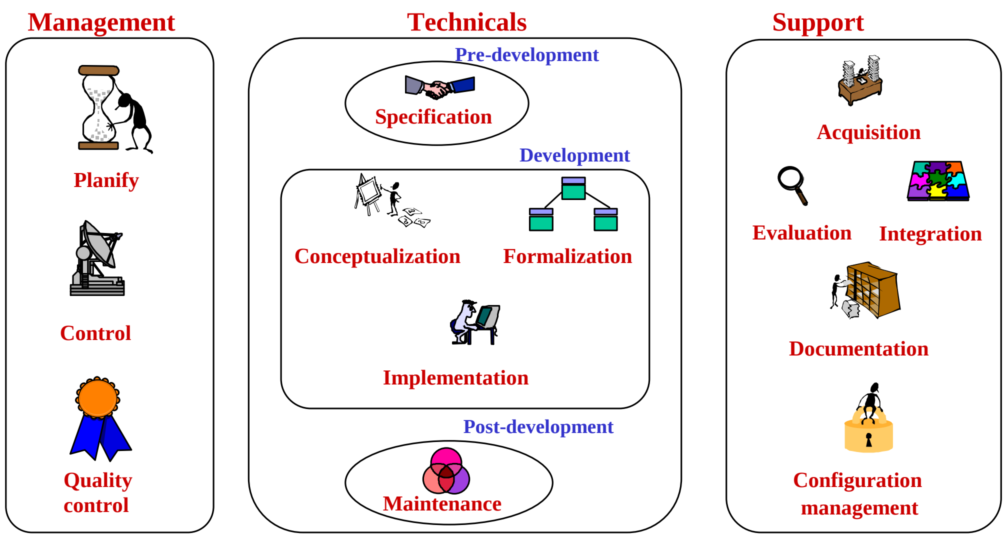

Figure 7.1: Ontology Development Process [19]

The previous ontology development process identifies the activities to be performed. It does not say nothing about how the must be scheduled. This is determined by the other part of the methodology, the ontology life cycle, that establishes the stages through which the ontology moves during its life time and the activities to be performed in each stage.

7.2.2 Ontology Life Cycle

The ontology life cycle schedules the ontology development activities detailed previously, although not all of them are currently considered by the Methontology life cycle. The life cycle is cyclic, based on evolving prototypes [35]. It allows an incremental development of the ontology that enables earlier validation and readjustment. Each cycle starts with the *scheduling* activity that identifies the tasks to be performed, their arrangement, their temporal extent and the resources they need. After that the development activities are engaged, starting with *specification*. Simultaneously, the management activities, *control* and *quality assurance*, and the support activities, *knowledge acquisition*, *integration*, *evaluation*, *documentation* and *configuration management*, are launched. They take place in parallel with the development activities.

Each cycle, the current prototype ontology moves along the development activities, from *specification* through *conceptualisation*, *formalisation* and *implementation* until *maintenance*, although it is not necessary to pass through all them. Eventually, the prototype might be mature enough for evaluation purposes and a new cycle can be engaged considering the conclusions from this evaluation. If a development cycle is completed, these are the steps that are performed:

1. To specify the prototype.
2. To build a conceptual model from pieces provided by the knowledge acquisition activity, which is mainly run during the conceptualisation.
3. To formalise the conceptual model.
4. To implement the formalised conceptual model. This can be automatic if the formalisation can be translated automatically to an ontology implementation language.
5. To maintain the resulting ontology, which might lead to a new development cycle if unsatisfied or new requirements are detected.

As it has been said and it is shown in Figure 7.2, the activities in the management and support processes take place simultaneously with the development activities. The efforts applied to the support activities are not uniform along the life cycle. *Knowledge acquisition*, *integration* and *evaluation* are greater during ontology *conceptualisation*. This is due to most knowledge is acquired at the beginning of the development, ontologies are integrated at the conceptual level before implementation and it is better to accurately evaluate the conceptualisation as earlier as possible in order to avoid propagating errors.

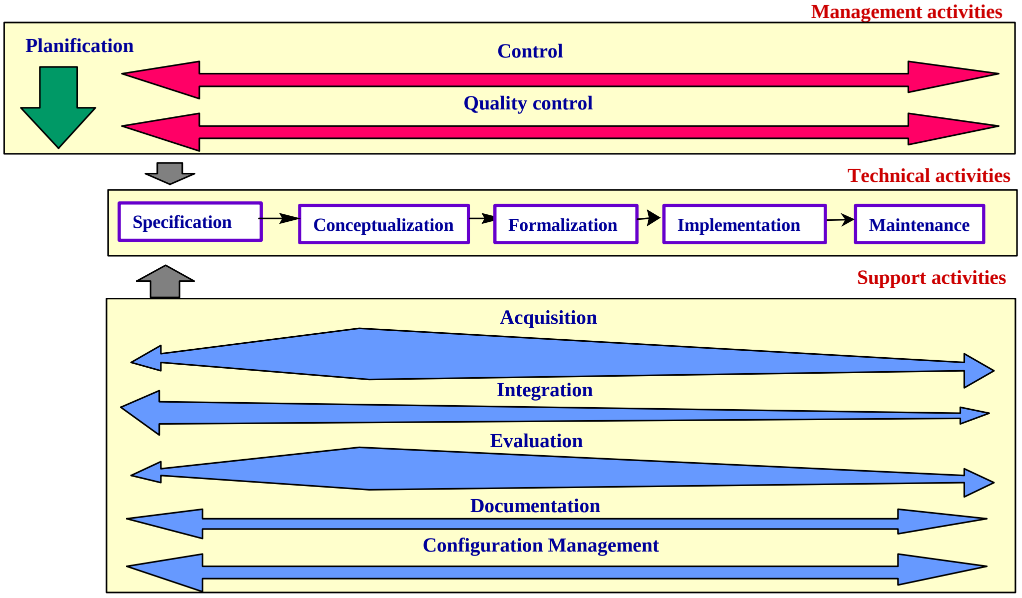

Figure 7.2: Methontology ontology development process life cycle [19]

It is important to note that all the relationships between activities detailed until this point are intra-dependencies, i.e. they are relationships between activities from the same ontology development process. Intra-dependencies define the ontology life cycle. Methontology considers also that activities for the development of an ontology may involve activities for other ontologies already built or under construction. These are called inter-dependencies and defined crossed life cycles of ontologies. They are necessary, for instance, because it is usually necessary to perform some changes before an ontology is integrated with the ontology currently under development.

Now, there are more detailed descriptions of the activities scheduled by the Methontology ontology life cycle. They are organised in the management, development and support processes. The pre-development activities (*environment* and *feasibility studies*) and the post-development activity *use* are not detailed as they are not included in the current life cycle. Moreover, the maintenance activity is moved to the development process.

Management Process

The management process activities are responsible for the project management issues [98].

Scheduling is the first activity of the ontology life cycle. The objective is to plan the main tasks to be done, how they will be arranged and the required resources, i.e. people, software and hardware.

Control is performed along the whole ontology life cycle in order to survey that there are not undesired deviations from the initial schedule.

Quality is responsible for checking that the quality of each methodology output (ontology, software and documentation) is assured.

Development Process

The development process includes all the activities that produce the successive prototype refinement stages towards the desired ontology[34]. The process starts with specification that produces an informal output that then evolves increasing its level of formality, as it passes through the different activities, towards the final computable model, which can be directly understood by the machine.

Specification The specification establishes the ontology purpose and scope. Why the ontology is being built, what are the intended uses and end-users [50]. The specification can be informal, in natural language, or formal, e.g. using a set of competence questions [113].

Conceptualisation The objective of this activity is to organize and structure the knowledge acquired during knowledge acquisition using external representations that are independent of the knowledge representation and implementation paradigms in which the ontology will be formalised and implemented next. An informally perceived view of a domain is converted into a semi-formal model using intermediate representations based on tabular and graph notations. These intermediate representations (concept, attribute, relation, axiom and rule) are valuable because they can be understood by domain experts and ontology developers. Therefore, they bridge the gap between people's domain perception and ontology implementation languages.

In order to build a consistent and complete conceptual model, the conceptualisation activity defines a set of tasks that should be executed in succession. These tasks increase, step by step, the complexity of the intermediate representations used to build the conceptual model. This way, it is easier to ensure a consistent and complete conceptual model:

1. First it is necessary to build a **glossary of terms** to be included on the ontology, their natural language definition and their synonyms and acronyms. Terms are identified following a middle-out strategy. The core of basic terms are identified first and then they are specialised and generalised as required. This strategy provides a balanced set of terms because detail only arises as necessary and higher level categories are built naturally.

2. Then, the terms are classified into one or more **taxonomies of concepts**, where a concept is an abstraction for one or more terms. The subclass of taxonomic relation is used, where: C subclass of D $\iff$ $\forall$i instance of C $\rightarrow$ i instance of D.

3. **Binary relations** are used to define the ad hoc relations between concepts of the ontology and also with concepts of other ontologies. Relations are determined by their name and the source and target concepts.

4. The **concept dictionary** is built. It describes each concept by stating the relations that have it as their domain and the concept instance and class attributes. Class **attributes** have the same value for all instances of a the concept, while instance attributes have different values for each instance of the concept. Moreover, it might be necessary now to define some concept **instances**, although it is more usual to create them during ontology use, after its construction.

5. The concept dictionary is detailed. For each relation, it is specified its cardinality, inverse relation and mathematical properties (symmetric, transitive, functional,...). Instance and class attributes are also described in terms of their concept, value type, measurement unit, range, cardinality, value and related axioms and rules that infer the value of this attribute or use it to infer other attributes. Moreover, there is a **constants** table that defines immutable aspects of the domain of knowledge.

6. Once concepts, taxonomies, attributes and relations have been defined, formal axioms and rules are used for constraint checking and for inferring values for attributes. **Axioms** are logical expressions that are always true and are normally used to specify constraints. They are defined informally in textual form and formally in first order logic, see the Logic section 4.8.7. Moreover, all the concepts, relations and attributes used in the definitions are highlighted. **Rules** are generally used to infer knowledge in the ontology, such as attribute values, relation instances, etc. Rules are also defined informally and formally and the related concepts, relations and attributes are highlighted. The "if conditions then consequent" rule template determines rules definitions. In order to avoid inference complexity problems, it is usual to

restrict the conditions to a conjunct of atoms and the consequent to a single atom. These restrictions might be relaxed if more complex inferences are needed.

Formalisation The goal of this activity is to formalise the conceptual model. There are ontology development tools that automatically implement the conceptual model into several ontology languages using translators. Therefore, formalisation is not a mandatory activity.

Implementation This activity builds computable models using ontology implementation languages. There are many ontology languages and they do not have the same expressiveness nor do they reason the same way.

Maintenance This activity updates and corrects the ontology if needed due to the necessities of the current development process or other processes that reuse this ontology in order to build other ontologies or applications.

Support Process

The support activities are performed in parallel with the development-oriented activities.

Knowledge Acquisition First of all, the source knowledge must be captured using knowledge elicitation techniques [113]. The sources of knowledge are listed giving a description and specifying the elicitation techniques used in each case. The techniques used to extract knowledge from sources can be partially automatic by means of natural language analysis and machine learning techniques [34, 52].

Evaluation The evaluation activity judges the developed ontologies, software and documentation against a frame of reference. Ontologies should be evaluated before they are used or reused. There are two kinds of evaluation, the technical one, which is carried out by developers, and users evaluation.

Ontology evaluation includes [1]:

- Ontology **verification** refers to building the ontology correctly, that is, ensuring that its definitions implement correctly the requirements or function correctly in the real world.

- Ontology **validation** refers to whether the ontology definitions really model the real world for which the ontology was created.
- Ontology **assessment** is focused on judging the ontology from the user's point of view. Different types of users and applications require different means of assessing an ontology.

The criteria for ontology evaluation are:

- **Consistency**, which checks if all individual definitions are consistent and no contradictory knowledge can be inferred from other definitions and axioms. Some consistency problems are: circular definitions, common classes or instances in disjoint decompositions and partitions, external instances in exhaustive decompositions and partitions and semantic errors.
- **Completeness**. All that is supposed to be in the ontology is explicitly stated in it, or it can be inferred. Some common completeness errors are: incomplete concept classification, disjoint knowledge omission and exhaustive knowledge omission.
- **Conciseness**. An ontology is concise if it does not include unnecessary definitions, explicit redundancies between definitions do not exist and redundancies cannot be inferred. Some redundancies are: redundant subclass of or instance of relations and identical formal definitions of classes or instances.

One method for ontology validation is Ontoclean [54], which is detailed in the next section.

Integration, merging and alignment The integration activity is needed if other ontologies are reused [98]. There are to options when an ontology is integrated in the current ontological framework. First, there is ontology alignment that consists in establishing different kinds of mapping between the ontologies, hence preserving the original ontologies. Second, ontology merging that produces a new ontology from the combination of the input ontologies.

Documentation Documentation details each completed stage and product.

Configuration Management Configuration management records ontologies, software and documentation versions in order to control changes [98].

7.3 XML Semantics Reuse

As it has been pointed out in the Objectives chapter 6, the intention is to take profit from previous work in the RELs field. Almost all existing REL initiatives are based on XML. XML schemas are used to define REL grammars and they are the source from which the semantics they capture implicitly are going to be formalised and made explicit. A generic methodology for XML semantics reuse has been employed [42]. It is based on mapping from XML Schema constructs to the OWL ones that are semantically more appropriate. The previous mapping is complemented with a XML instance metadata to RDF instance metadata mapping. The latter makes possible to take existing XML metadata to the Semantic Web space.

There are many attempts to make XML metadata semantics explicit. Usually, they translate it to Semantic Web languages that facilitate the forma-lisation. Some of them just model the XML tree using the RDF primitives [71]. Others concentrate on modelling the knowledge implicit in XML languages definitions, i.e. DTDs or the XML Schemas, using web ontology lan-guages [3, 21]. Finally, there are attempts to encode XML semantics inte-grating RDF into XML documents [76, 90].

However, none of them facilitates an extensive transfer of XML metadata to the Semantic Web in a general and transparent way. Their main problem is that the XML Schema implicit semantics are not made explicit when XML metadata instantiating this schemas is mapped. Therefore, they do not take profit from the XML semantics and produce RDF metadata almost as semantics-blind as the original XML. Alternatively, they capture this semantics but they use additional ad-hoc semantic constructs that produce less transparent metadata.

Therefore, we have chosen the XML Semantics Reuse methodology that combines a XML Schema to web ontology mapping, called XSD2OWL, with a transparent mapping from XML to RDF, XML2RDF. The ontologies generated by XSD2OWL are used during the XML to RDF mapping in order to generate semantic metadata that makes XML Schema semantics explicit. Both steps are detailed in the next subsections.

7.3.1 XML Schema to OWL

As we have said, XML Schemas define some simple semantics. For instance, the *substitutionGroup* relations among elements and the *extension/restriction base* ones among *complexTypes* encode generalisation hierarchies. The XML Schema to OWL mapping is responsible for capturing the schema implicit semantics. This semantics are determined by the combination of XML Schema constructs.

The mapping is based on translating this constructs to the OWL ones that best capture their semantics. These translations are detailed in Table 7.1.

In each row there is a mapping. If there is more than one line, there are indeed two mapping but very related. In the first column there are the XML constructs detailed using a XPath syntax [103]. The second column contains the OWL constructs to which the corresponding XML Schema construct is mapped.

In the case of *elements* and *attributes*, the possible OWL constructs are *rdf:Property*, *owl:DatatypeProperty* and *owl:ObjectProperty*. *owl:DatatypeProperty* is used for all *attributes* and those elements that have a *simpleType* as value, i.e. a string, integer, etc. value. *owl:ObjectProperty* is used for the *elements* that have a *complexType* as value. Finally, it is necessary to use *rdf:Property* for those *elements* that may have both a *simpleType* or *complexType* value, as this is possible with XML Schema but it is not recommended in OWL.

For the rest of the mappings that have more than one line in its second column, as they also have more than one line in its first column, there are more than one mapping per row and each line corresponds to one mapping. The third column points out informally in which respect the XML Schema and OWL construct for the given mapping are related.

XML Schema	**OWL**	**Shared informal semantics**
element | attribute	rdf:Property owl:DatatypeProperty owl:ObjectProperty	Named relation between nodes or nodes and values
element@substitutionGroup	rdfs:subPropertyOf	Relation can appear in place of a more general one
element@type	rdfs:range	The relation range kind
complexType | group | attributeGroup	owl:Class	Relations and contextual restrictions package
complexType//element	owl:Restriction	Contextualised restriction of a relation
extension@base | restriction@base	rdfs:subClassOf	Package concretises the base package
@maxOccurs @minOccurs	owl:maxCardinality owl:minCardinality	Restrict the number of occurrences of a relation
sequence choice	owl:intersectionOf owl:unionOf	Combination of relations in a context

Table 7.1: XSD2OWL translations for the XML Schema constructs and shared semantics with OWL constructs

The XSD2OWL mapping is quite transparent and captures a great part XML Schema semantics.

The same names used for XML constructs are used for OWL ones, although in the new namespace defined for the ontology. Therefore, XSD and OWL constructs names are identical; this usually produces uppercase-named OWL properties because the corresponding element name is uppercase, although this is not the usual convention in OWL. Moreover, it also possible to have anonymous XML constructs, concretely *complexTypes* defined implicitly inside a *element* definition. In this case, the OWL constructs are named with *element* name concatenated with the "Range" word.

The only caveats are the implicit order conveyed by *sequence* and the exclusivity of *choice*. For the first problem, *owl:intersectionOf* does not retain its operands order, there is no clear solution that retains the great level of transparency that has been achieved. The use of RDF Lists might impose order but introduces ad-hoc constructs not present in the original metadata. Moreover, as it has been demonstrated in practise, the elements' ordering does not contribute much from a semantic point of view. For the second problem, *owl:unionOf* is an inclusive union, the solution is to use the disjointness OWL construct, *owl:disjointWith*, between all union operands in order to make it exclusive.

For the predefined *simpleTypes* that are included in the current OWL specification, i.e. *datatypes* like *xsd:string* or *xsd:boolean*, the mapping is direct. For the user defined *simpleTypes*, as there is not a standard method for custom datatypes in OWL, all user defined *simpleTypes* are mapped to *xsd:string* in order to keep their lexical values intact. Although this causes a loose of semantic information in the resulting OWL ontology, it is still possible to validate instance XML metadata against the original XML Schema prior to mapping it to RDF. Consequently, this lack in the OWL ontology can be overcome and the final RDF metadata is consistent with the XML Schema user defined *simpleTypes*.

Finally, some post-mapping adjustments may be necessary in order to solve name collisions between an OWL class and a RDF property. This is due to the fact that XML has independent name domains for *complexTypes* and *elements* while OWL has a unique name domain for all constructs. Moreover, the resulting OWL ontology is OWL-Full because the XSD2OWL translator has employed *rdf:Property* for those *elements* that have both data-type and object-type ranges.

7.3.2 XML to RDF

Once XML Schemas are available as mapped OWL ontologies, it is also possible to map the XML metadata that instantiates them. The intention is to produce RDF metadata as transparently as possible. Therefore, a structure-mapping approach has been selected [71]. It is also possible to take a model-mapping approach [109]. XML model-mapping is based on representing the XML information set using semantic tools. This approach is better when XML metadata is semantically exploited for concrete purposes. However, when the objective is semantic metadata that can be easily integrated, it is better to take a more transparent approach.

Transparency is achieved in structure-mapping models because they only try to represent the XML metadata structure, i.e. a tree, using RDF. The RDF model is based on the graph so it is easy to model a tree using it. Moreover, we do not need to worry about the semantics loose produced by structure-mapping. We have formalised the underlying semantics into the corresponding ontologies and we will attach them to RDF metadata using the instantiation relation *rdf:type*.

The structure-mapping is based on translating XML metadata instances to RDF ones that instantiate the corresponding constructs in OWL. The more basic translation is between relation instances, from *xsd:elements* and *xsd:attributes* to *rdf:Properties*. Concretely, *owl:ObjectProperties* for node to node relations and *owl:DatatypeProperties* for node to values relations. However, in some cases, it would be necessary to use *rdf:Properties* for *xsd:elements* that have both data type and object type values.

Values are kept during the translation as simple types and RDF blank nodes are introduced in the RDF model in order to serve as source and destination for properties. They will remain blank for the moment until they are enriched with semantic information, as it is shown in Figure 7.3.

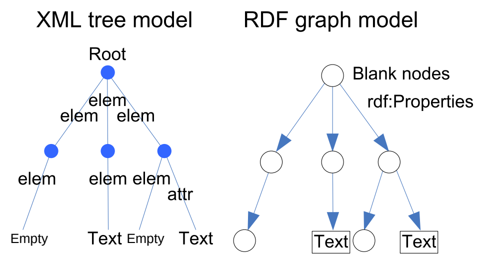

Figure 7.3: XML vs. RDF Models

The resulting RDF graph model contains all that we can obtain from the XML tree. It is already semantically enriched thanks to the *rdf:type* relation that connects each RDF properties to the *owl:ObjectProperty* or *owl:DatatypeProperty* it instantiates. It can be enriched further if the blank nodes are related to the *owl:Class* that defines the package of properties and associated restrictions they contain, i.e. the corresponding *xsd:complexType*. This semantic decoration of the graph is formalised using *rdf:type* relations from blank nodes to the corresponding OWL classes.

At this point we have obtained a semantics-enabled representation of the input metadata. The instantiation relations can now be used to apply OWL semantics to metadata. Therefore, the semantics derived from further enrichments of the ontologies, e.g. integration links between different ontologies or semantic rules, are automatically propagated to instance metadata thanks to inference.

These mappings have been validated in different ways. First, OWL validators have beem used in order to check the resulting OWL ontologies. Moreover, the two mappings have been tested in conjunction. Testing XML instances have been mapped to RDF, guided by the corresponding OWL ontologies from the used XML Schemas, and then back to XML. Then, the original and derived XML instances have been compared using their canonical version in order to correct mapping problems.

Part III

Contribution

Chapter 8

Specification

8.1 Summary

In the specification, we are going to analyse the subject domain, i.e. copyright. First of all, it is situated in the broader context of intellectual property (IP) at the international level, which is defined by the international agreements managed by the World Intellectual Property Organisation (WIPO). This section shows that IP is divided into industrial property and copyright. As our intention is to deal with literary, artistic and scientific works, the focus is going to be placed on the latter part.

We will not deal with inventions, which are, on the contrary to scientific discoveries, basically technical. Patents govern them. We will not deal with trademarks and industrial designs either. Therefore, we are going to deal just with copyright. It is important to note that we understand copyright from the wider definition hold by the WIPO. Traditionally, copyright has been associated with just the economic rights, also know as exploitation rights. This is common in Anglo-Saxon legal traditions like those in the United Kingdom or the USA. However, in addition to economic rights, and as WIPO does, we are also considering moral rights.

Once the focus is placed on copyright, this specification concentrates on the analysis of its main concepts. First, the concept of "Work" is detailed. What are considered works? What requirements must be satisfied in order to have a copyrighted work? What kinds of copyrighted works are there? Then, copyright is divided into the different rights that compose it. They govern very specific kinds of actions that can be carried out on works. The rights related to copyright are also described, i.e. those corresponding to performers, producers and broadcasters. To conclude, the limitations and exceptions of copyright are shown.

8.2 Intellectual Property

Intellectual property, very broadly, means the legal rights that result from intellectual activity in the industrial, scientific, literary and artistic fields. Intellectual property includes rights related to:

- Literary, artistic and scientific works.
- Performances of performing artists, phonograms and broadcasts.
- Inventions in all fields of human endeavour.
- Industrial designs.
- Trademarks, service marks, commercial names and designations.

Countries have laws to protect intellectual property for two main reasons. One is to give statutory expression to the moral and economic rights of creators in their creations and the rights of the public in access to those creations. The second is to promote, as a deliberate act of Government policy, creativity and the dissemination and application of its results and to encourage fair-trading, which would contribute to economic and social development.

Generally speaking, intellectual property law aims at safeguarding creators and other producers of intellectual goods and services by granting them certain time-limited rights to control the use made of their productions. Those rights do not apply to the physical object in which the creation may be embodied but instead to the intellectual creation as such.

Intellectual property is traditionally divided into two categories [122]:

- **Industrial property**, which includes inventions (patents), trademarks, industrial designs, and geographic indications of source.
- **Copyright**, which includes literary and artistic works such as novels, poems and plays, films, musical works, artistic works such as drawings, paintings, photographs and sculptures, and architectural designs. Rights related to copyright include those of performing artists in their performances, producers of phonograms in their recordings, and those of broadcasters in their radio and television programs.

Industrial property covers inventions and industrial designs. Simply stated, inventions are new solutions to technical problems and industrial designs are aesthetic creations determining the appearance of industrial products. In addition, industrial property includes trademarks, service marks, commercial names and designations, including indications of source and appellations of origin, and protection against unfair competition. Here, the aspect of intellectual creations is less prominent, but what counts here is that the object of industrial property typically consists of signs transmitting information to consumers, in particular as regards products and services offered on the market, and that the protection is directed against unauthorized use of such signs which is likely to mislead consumers, and misleading practices in general.

On the other hand, copyright, as understood by the WIPO deals with all the aspects of literary, artistic and scientific works we are interested in, i.e. their economic exploitation but also the moral rights of the author. Traditionally, copyright has been associated with just the economic rights, also know as exploitation rights. Another term that we will also use to refer to the union of economic and moral rights is author rights. This is the more concrete term although it is common just in continental Europe legal tradition.

The WIPO is promoting the introduction of moral rights in all legal systems in order to harmonise Intellectual Property worldwide and the WIPO definition of copyright does already include them. Therefore, copyright is the focus of this work and is analysed in the next section with greater detail. Figure 8.1 shows a summary of all these terms and how they are related to more general terms.

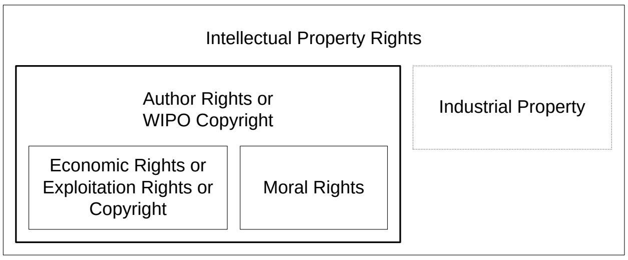

Figure 8.1: Organisation of the terms related to Intellectual Property

8.3 Copyright

Copyright law is a branch of that part of the law that deals with the rights of intellectual creators [9]. Copyright law deals with particular forms of creativity, concerned primarily with mass communication. It is concerned also with virtually all forms and methods of public communication, not only printed publications but also such matters as sound and television broadcasting, films for public exhibition in cinemas, etc. and even computerized systems for the storage and retrieval of information.

Copyright deals with the rights of intellectual creators in their creation. Most works, for example books, paintings or drawings, exist only once they are embodied in a physical object. But some of them exist without embodiment in a physical object. For example music or poems are works even if they are not, or even before they are, written down by a musical notation or words. However, there are some legal systems that do not protect the copyright of works that have not been fixed in some form.

Copyright law protects only the form of expression of ideas, not the ideas themselves. The creativity protected by copyright law is creativity in the choice and arrangement of words, musical notes, colours, shapes and so on. Copyright law protects the owner of rights in artistic works against those who take and use the form in which the original work was expressed by the author.

At the international level, the Berne Convention confers the economic and moral rights for the protection of literary and artistic works, commonly known as the Berne Convention. This Convention, which was adopted in 1886, has been revised several times to take into account the impact of new technology on the level of protection that it provides. It is administered by the World Intellectual Property Organization (WIPO), one of the specialized international agencies of the United Nations system. The Treaties section 8.3.5 contains a full list of all the international regulations managed by the WIPO.

Legislation provides protection not only for creators of intellectual works but also for the auxiliaries that help in the dissemination of such works. This auxiliaries protected by legislation are performers, producers and broadcasters. The owner of copyright in a work is generally, at least in the first instance, the person who created the work, that is to say, the author of the work. There can be exceptions to this general principle and national laws regulate such exceptions. For example, some national laws provide that, when a work an author employed for the purpose of creating that work, then the employer, not the author, is the owner of the copyright in the work.

From this initial situation, copyright, with the exception of moral rights, may be assigned. This means that the owner of the copyright transfers it to another person or entity, which becomes the owner of the copyright. In some other countries, an assignment of copyright is not legally possible. However, licensing achieves almost the same practical effect than assignment.

Licensing means that the owner of the copyright remains the owner but authorizes someone else to exercise all or some of his rights, usually subject to some limitations. When such authorization or license extends to the full period of copyright and when such authorization or license extends to all the rights, the licensee is for all practical purposes in the same position as an owner of copyright. However, this is just from the economic rights point of view. Moral rights cannot be licensed neither be transferred.

8.3.1 Work

The subject matter of copyright protection includes every production in the literary, scientific and artistic domain, whatever the mode or form of expression. For a work to enjoy copyright protection, however, it must be an original creation. The ideas in the work do not need to be new but the form, literary or artistic, in which they are expressed, must be an original creation of the author. And, finally, protection is independent of the quality or the value attaching to the work. It will be protected whether it be considered, according to taste, a good or a bad literary or musical work and even of the purpose for which it is intended, because the use to which a work may be put has nothing to do with its protection.

Works eligible for copyright protection are, as a rule, all original intellectual creations. A non-exhaustive, illustrative enumeration of these is contained in national copyright laws. To be protected by copyright law, an author's works must originate from him; they must have their origin in the labour of the author. But it is not necessary, to qualify for copyright protection, that works should pass a test of imaginativeness, of inventiveness. The work is protected irrespective of the quality thereof and also when it has little in common with literature, art or science, such as purely technical guides or engineering drawings, or even maps. Exceptions to the general rule are made in copyright laws by specific enumeration; thus laws and official decisions or mere news of the day are generally excluded from copyright protection.

Practically all copyright law systems provide for the protection of the following types of work:

- **Literary works***:* novels, short stories, poems, dramatic works and any other writings, irrespective of their content (fiction or non-fiction), length, purpose (amusement, education, information, advertisement, propaganda, etc.), form (handwritten, typed, printed; book, pamphlet, single sheet, newspaper, magazine); whether published or unpublished; in most countries "oral works", that is, works not reduced to writing, are also protected by the copyright law.
- **Musical works***:* whether serious or light; songs, choruses, operas, musicals, operettas; if for instructions, whether for one instrument (solos), a few instruments (sonatas, chamber music, etc.), or many (bands, orchestras).
- **Dramatic**, **pantomimes** and **choreographic works***:* including any accompanying music.
- **Artistic works***:* whether two-dimensional (drawings, paintings, etchings, lithographs, etc.) or three-dimensional (sculptures, architectural works), irrespective of content (representational or abstract) and destination ("pure" art, for advertisement, etc.);

- **Maps**, **technical drawings** and **architectural works**.
- **Photographic works**: irrespective of the subject matter (portraits, landscapes, current events, etc.) and the purpose for which they are made.
- **Motion pictures**: whether silent or with a soundtrack, and irrespective of their purpose (theatrical exhibition, television broadcasting, etc.), their genre (film dramas, documentaries, newsreels, etc.), length, method employed (filming "live," cartoons, etc.), or technical process used (pictures on transparent film, videotapes, DVDs, etc.).
- **Computer programs**: either as a literary work or independently depending on the concrete legal system.

Please note that mere ownership of the material support of a copyrighted work, i.e. a Compact Disc or a painting, does not give you the automatic right to copy part or all of that work. If you are not the copyright holder, you are ordinarily limited to making one archival copy (reserved for your own use in case the original becomes damaged). Even where you make the outright purchase of an original work of art, the original artist may retain certain rights in the manner in which the artwork is displayed, and may through a contract of sale retain the right to reproduce the work ownership of the original work alone will not necessarily entitle you to make or sell copies.

Requirements

As it has been said, the first requirement in order to be a copyrighted work is originality, which is detailed below. Moreover, there are some kinds of works that are not subject to copyright:

- **Ideas, procedures, methods, systems, processes, concepts, principles, discoveries, or devices**: they are not subject to copyright, as distinguished from a description, explanation, or illustration of those ideas, systems and processes, which are protected by copyright.
- **Utilitarian works**: a purely utilitarian work is not subject to copyright, as "utility" is not considered to be expression. Please note, however, that creative aspects of a utilitarian work remain subject to copyright. A belt buckle is a utilitarian item, but decoration on the belt buckle may still be copyrighted. Please note also that it may be possible to secure patent protection for an original utilitarian design.
- **Facts**: Facts are not subject to copyright, no matter how difficult it may be for the person who has published a factual work to find and present those facts. Similarly, lists of ingredients are not subject to copyright. Please remember that the author's unique presentation of facts is still subject to copyright, even though the facts themselves are open to public use.

A part from these conditions, in some countries there is also the need for the work to be fixed in some way in order to be protected. No other actions are required for copyright protection. There is no need to file an application for copyright protection, or to even place a copyright notice on a work. These additional steps, often referred to as "formalities", were previously required to secure copyright protection. Under the current law, the formalities of registration and notice now merely serve as recommended steps to expand the protection provided by copyright.

Originality For a work to be protected by copyright law, it must be "original". The ideas in the work do not need to be new but the form, literary or artistic, in which they are expressed, must be an original creation of the author. However, the amount of originality required is extremely small. The work cannot be a mere mechanical reproduction of a previous work, nor can the work consist of only a few words or a short phrase. In addition, if the work is a compilation, the compilation must involve some originality beyond mere alphabetic sorting of all available works. Beyond that, almost any work that is created by an author will meet the originality requirement, although it might be considered a derivation from a previous work.

The author of the work must be a human being. The works performed by machines are not protected. In these cases, what is protected instead is the mechanism or procedure that generates the work. For instance, a computer program, which generates drawings, music or translations, will be protected by copyright and the works that it generates will not. Photos are also produced by machines, however what is valued in this case is the contribution of the photographer in order to choose the framing, exposure, etc.

The definition of originality is another important point. Usually, it is understood subjectively, i.e. the work is original in the sense that it is not a copy of a previous work and at least can be considered a derivation as it contributes some original work. On the contrary, objective originality, i.e. to create something new, is usually employed in the patents domain. Moreover the level of originality required is dependent on the field of the contribution. For instance there is not the same originality requirement for major works, like books or songs, than for minor works, like flyers or slogans. Correspondingly, the level of protection against similar creations depends on the creativity contribution of the work. When the contribution is small it is easier that similar works are considered distinct works.

Fixation In some countries, works that have not been fixed in a tangible form of expression are not subject to copyright. For example, a poem, a dance work that has never been notated or recorded or a speech or performance that was not written and was not recorded, may not be subject to copyright.

For instance the U.S.A. as stated in the U.S. Copyright Act, in order for a work to be protectable, it must be fixed in a tangible medium of expression. A work is considered fixed when it is stored on some medium in which it can be perceived, reproduced, or otherwise communicated. For example, a song is considered fixed when it is written down on paper. The paper is the medium on which the song can be perceived, reproduced and communicated.

It is not necessary that the medium be such that a human can directly perceive the work from it, as long as a machine can perceive the work. Thus, the song is also fixed the moment the author records it onto a cassette tape. Similarly, a computer program is fixed when stored on a computer hard drive. In fact, courts have even held that a computer program is fixed when it exists in the RAM of a computer. This is true even though this "fixation" is temporary, and will disappear once power is removed from the computer.

However, this is not the general case. There are other countries that extend copyright protection to unfixed works, e.g. poems, music, dance works, speeches, etc. In any case, it is very complicated to protect a work that has never been fixed. It is very difficult to demonstrate authorship if there is not material evidence.

Copyright Notice From the moment an original work is fixed in a tangible medium of expression, copyright applies whether or not there is a notice of copyright affixed to the work. A created work is considered protected by copyright as soon as it exists. According to the Berne Convention for the Protection of Literary and Artistic Works, literary and artistic works are protected without any formalities in the countries party to that Convention. Thus, WIPO does not offer any kind of copyright registration system.

However, a copyright notice helps protect an original work by protecting against a claim of innocent infringement, and by helping people who wish to license the work to find and contact the author. The notice should be affixed in such a way as to give reasonable notice of the claim of copyright. A proper copyright notice includes three elements:

1. The symbol © or the word "copyright".
2. The first year in which copies of the work were published or distributed to others.
3. The name of the copyright owner.

In order to maximize your protections under international conventions, you should always utilize the symbol © in your copyright notices (or the symbol for sound recordings), and should also include the phrase: "All rights reserved". For example: "© 2005 Roberto García González, All Rights Reserved".

Moreover, many countries have a national copyright office and some national laws allow for registration of works for the purposes of, for example, identifying and distinguishing titles of works. In certain countries, registration can also serve as evidence in a court of law with reference to disputes relating to copyright.

Kinds of Works

Compilation A compilation is a work that is formed by the collection and assembling of pre-existing materials or of data (databases) that are selected in such a way that the resulting work as a whole constitutes an original work of authorship. An example of a compilation would be a collection of the most influential plays of the Eighteenth Century. The individual plays themselves would not be subject to copyright protection, since the copyright would have expired. However, the selection of the plays, as well as their order, involves enough original and creative expression to be protected by copyright. Therefore, the grouping of plays is protected by the copyright in the compilation even though each individual play is not protected.

A grouping of facts is also protected as a compilation, assuming the grouping contains enough original expression to merit protection. An example of a protectable grouping of facts would be a web site containing links to other web sites. Each link consists merely of factual information, namely that a particular web site can be found at a particular URL location. Thus, there is no copyright protection for the links. Although the individual links can be copied and placed unto another web site, if the entire list (or a substantial portion) of the list were copied, the copyright in the compilation would be infringed. The creative, original expression that is being protected is the sorting, selecting, and grouping of all the selected web sites into the list found on this web page.

The white pages telephone directory is an example of an unprotected grouping of facts. The individual facts (name, address, and telephone number) are not protectable under the copyright law. In addition, the compilation in this case consisted solely of gathering all available telephone numbers in a particular area and sorting them alphabetically. The U.S. Supreme Court has held that this minimal level of selecting and arrangement does not involve enough originality to be protected by copyright.

Collective The initiative to create a collective work is carried out by a special individual. This individual, a juridical or natural person, also coordinates the creative process, divulgates the work and usually finances it. This person is considered the author with all the corresponding rights. The work is built from the contributions of different authors, who are coordinated by the promoter, which are combined in a unique an autonomous work.

Copyright in each separate contribution to a collective work is distinct from copyright in the collective work as a whole, which is held by author of the contribution. In the absence of an express transfer of the copyright or of any rights under it, the owner of copyright in the collective work is presumed to have acquired only the privilege of reproducing and distributing the contribution as part of that particular collective work, any revision of that collective work, and any later collective work in the same series.

Hired In the case of works made for hire, where an artist has created the work as an employee, the employer, and not the employee, is considered to be the author and copyright holder.

Collaboration Two or more authors, who collaborate directly or indirectly, create a collaborative work. They will share the copyright on the resulting work unless there is an agreement to the contrary.

Derivation This kind of works is based in pre-existent ones without the collaboration of their original authors. Their rights must be considered and their authorisation is required. Derivations, although based in previous ones, must be original as any other work, i.e. they contribute something and a new work can be identified. If the changes are not substantial, the result is a reproduction.

New derivation is dependent on the pre-existent work because it maintains some of their characteristic features. This is different to say that a work is inspired in previous works, which happens always consciously or not and does not have legal implications. Inspiration, like ideas, is totally free.

Therefore, the originality of derivations is based on the adaptation of an original work or its translation to a different language. Examples of derivations are: the adaptation of a dramatic work to novel or the translation of a film. The adaptation of a literary work to music is not considered derivation, it is considered inspiration.

The author of a derived work enjoys the same rights than any other author while the rights situation of the original work is not affected in any way. The rights holder of the latter may continue authorising transformations while they are also original with regard to other previously derived works.

However, in order to realise and commercially exploit the derived work an authorisation of the original rights holder is required. Moreover, there might be a chain of derivations that implies a chain of authorisations from the rights holders of the preceding works each time a new derivation is intended. This means that the authorisation to transform a work and to exploit the derived work does not imply the consent to new transformations from the derived work. The full chain of authorisations must be followed.

8.3.2 Rights

The owner of copyright in a protected work may use the work as he wishes, but not without regard to the legally recognized rights and interests of others, and may exclude others from using it without his authorization.

Therefore, the rights bestowed by law on the owner of copyright in a protected work are frequently described as exclusive rights to authorize others to use the protected work. The original authors of works protected by copyright also have moral rights, in addition to their exclusive rights of an economic character.

What is meant by using a work protected by copyright? Most copyright laws define the acts in relation to a work which cannot be performed by persons other than the copyright owner without the authorization of the copyright owner. Such acts, requiring the authorization of the copyright owner, normally are like the following:

- Copying or reproducing the work.
- Performing the work in public.
- Making a sound recording of the work.
- Making a motion picture of the work.
- Making the work available for the public.
- Broadcasting the work.
- Translating the work.
- Adapting the work.
- Etc.

Specific rights in copyright govern these acts. They are shown as groups of related rights in Figure 8.2 and detailed in the next subsections.

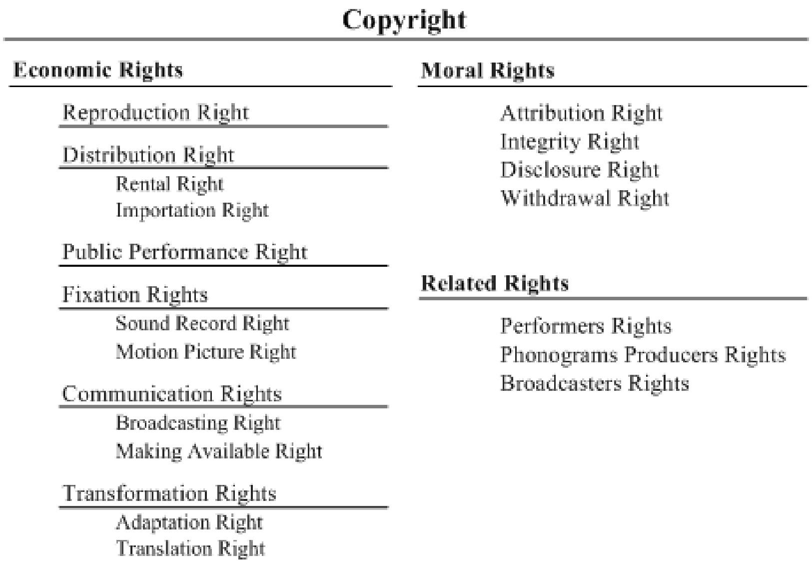

Figure 8.2: Rights in copyright from the international law point of view (WIPO)

Reproduction Right

The right of the owner of copyright to prevent others from making copies of his works is the most basic right under copyright. For example, the making of copies of a protected work is the act performed by a publisher who wishes to distribute copies of a text-based work to the public, whether in the form of printed copies or digital media such as CD-ROMs. Likewise, the right of a phonogram producer to manufacture and distribute compact discs (CDs) containing recorded performances of musical works is based, in part, on the authorization given by the composers of such works to reproduce their compositions in the recording. Therefore, the right to control the act of reproduction is the legal basis for many forms of exploitation of protected works.

Other rights are recognized in national laws in order to ensure that the basic right of reproduction is respected. They are related to the exploitation of the resulting copies by distributing them to the public, renting them or importing them.

Distribution Right

This right authorises the distribution to the public of previously made copies of works incorporated in a tangible article. The right of distribution is usually subject to exhaustion upon first sale or other transfer of ownership of a particular copy. This means that, after the copyright owner has sold or otherwise transferred ownership of a particular copy of a work, the owner of that copy may dispose of it without the copyright owner's further permission, for example, by giving it away or even by reselling it.

Rental Right The right to authorize rental of copies of works is justified because technological advances have made it very easy to copy these types of works. Experience in some countries has shown that copies were made by customers of rental shops, and therefore, that the right to control rental practices was necessary in order to prevent abuse of the copyright owner's right of reproduction.

Importation Right Some copyright laws include a right to control importation of copies as a means of preventing erosion of the principle of territoriality of copyright. This is because the legitimate economic interests of the copyright owner would be endangered if he could not exercise the rights of reproduction and distribution on a territorial basis. It is important to note that, despite of globalisation and digitalisation, this has still meaning as reproduction right and its related rights refer to tangible copies of works.

Public Performance Right

Another act requiring authorization is the act of public performance. To perform a work means to recite, render, play, dance, or act it, either directly or by means of any device or process or, in the case of a motion picture or other audiovisual work, to show its images in any sequence or to make the sounds accompanying it audible. This right just considers public performances, i.e. performances before an audience.

It is important to note that the performance is considered public when it takes place at a place open to the public or at any place where a substantial number of persons outside of a normal circle of a family and its social acquaintances is involved. Therefore, it will not be considered public when it is performed in a strictly domestic domain.

The right to control this act of public performance is of interest not only to the owners of copyright in works originally designed for public performance, but when others may wish to arrange the public performance of works originally just intended to be used by being reproduced and published.

Examples of public performances are:

- The scenic play of dramatic works.

- The public performance of musical works.
- The projection or public exhibition of cinematographic and audiovisual works.
- Etc.

Fixation Rights

A work is considered fixed when it is stored on some medium in which it can be perceived, reproduced, or otherwise communicated. For example, a song is considered fixed when it is written down on paper. The paper is the medium on which the song can be perceived, reproduced and communicated. It is not necessary that the medium be such that a human can perceive the work, as long as the work can be perceived by a machine. Thus, the song is also fixed the moment the author records it onto a cassette tape. Similarly, a computer program is fixed when stored on a computer hard drive.

Sound Record Right This is the right that governs the act of making a sound recording of a work protected by copyright. Sound recordings can incorporate music alone, words alone or both music and words. The right to authorize the making of a sound recording belongs to the owner of the copyright in the music and also to the owner of the copyright in the words. If the two owners are different, then, in the case of a sound recording incorporating both music and words, the maker of the sound recording must obtain the authorization of both owners. Under the laws of some countries, the maker of a sound recording must also obtain the authorization of the performers who play the music and who sing or recite the words.

Motion Picture Right A motion picture is a visual recording, giving to viewers an impression of motion. In the technical language of copyright law it is often called a cinematographic work or an audiovisual work. A drama originally written for performance by performers to an immediately present audience, i.e. a live performance can be visually recorded and shown to audiences far larger in numbers than those who can be present at the live performance. Such audiences can see the motion picture far away from the place of live performance and at times much later than the live performance.

Communication Rights

This is the right to authorize any communication to the public of the originals or copies of works, including wire or wireless means and "the making available to the public of works in a way that the members of the public may access the work from a place and at a time individually chosen by them". The quoted expression covers in particular on-demand, interactive communication through the Internet. This right just covers all communication to the public not present at the place where the communication originates. This right should cover any such transmission or retransmission of a work to the public by wire or wireless means, including broadcasting.

Broadcasting Right When a work is broadcasted, a wireless signal is emitted into the air, which can be received by any person, within range of the signal, who possesses the equipment (radio or television receiver) necessary to convert the signal into sounds or sounds and images. When a work is communicated to the public by cable, a signal is diffused and only persons who possess the required equipment linked to the cables used to diffuse the signal can receive it.

The broadcasting and diffusion by cable of works protected by copyright have given rise to new problems resulting from technological advances, which have introduced changes in copyright law. The advances include the use of space satellites to extend the range of wireless signals, the increasing possibilities of linking radio and television receivers to signals diffused by cable, and the increasing use of equipment able to record sound and visual images, which are broadcast or diffused by cable. This has originated the creation of a related right associated to broadcasters, which is detailed in the Related Rights section 8.3.3.

Making Available Right Due to recent technological advances, among which the Internet is the more relevant one, copyright has also included a particular kind of communication where members of the public access works from a place and at a time individually chosen by them. This kind of actions, i.e. interactive on-demand transmissions, is common in recent communication mediums like the Internet or mobile communications networks.

Transformation Rights

The acts of translating or of adapting a work protected by copyright require the authorization of the copyright owner. Translations and adaptations are themselves works protected by copyright. Therefore, in order, for example, to reproduce and publish a translation or adaptation, the publisher must have the authorization both of the owner of the copyright in the original work and of the owner of copyright in the translation or adaptation.

Moreover, there can be chains of transformations that force a chain of authorisations from the owner of the copyright of the original work through all transformed works until the current transformation.

Translation Right To translate means the expression of a work in a language other than that of the original version.

Adaptation Right To Adapt is generally understood as the modification of a work from one type of work to another, for example adapting a novel so as to make a motion picture, or the modification of a work so as to make it suitable for different conditions of exploitation, for example adapting an instructional textbook originally prepared for higher education into an instructional textbook intended for students at a lower level.

Moral Rights

The Berne Convention requires member countries to grant to authors:

- **Attribution Right**: the right to claim authorship of the work;
- **Integrity Right**: the right to object to any distortion, mutilation or other modification of, or other derogatory action in relation to, the work which would be prejudicial to the author's honour or reputation.

These rights, which are generally known as the moral rights of authors, are required to be independent of the usual economic rights and to remain with the author even after he has transferred his economic rights.

There are countries where additional moral rights are also considered:

- **Disclosure Right**: exclusive right to disclose the work.
- **Withdrawal Right**: exclusive right to withdraw the work.

8.3.3 Related Rights

There exist rights related to, or "neighbouring on", copyright. These rights are generally referred to as "related rights" or "neighbouring rights" in an abbreviated expression. It is generally understood that there are three kinds of related rights:

- The rights of performing artists in their performances.
- The rights of producers of phonograms in their phonograms.
- The rights of broadcasting organizations in their radio and television programs.

Protection of those who assist intellectual creators to communicate their message and to disseminate their works to the public at large is attempted by means of related rights. A play needs to be presented on the stage; a song needs to be performed by artists, reproduced in the form of records or broadcast by means of radio facilities. All persons who make use of literary, artistic or scientific works in order to make them publicly accessible to others require their own protection against the illegal use of their contributions in the process of communicating the work to the public.

Several countries also grant a sort of moral right to performers to protect them against distortion of their performances and grant them the right to claim the mention of their name in connection with their performances. Some countries also protect the interests of broadcasting organizations by preventing the distribution on or from their territory of any program-carrying signal emitted to or passing through a satellite, by a distributor for whom the signal is not intended.

Performers Right

A publisher reproduces a manuscript in its final form without adding to the expression of the work as created by the author. The interests of book publishers are protected by means of copyright itself. The position is slightly different with regard to dramatic and musical works, pantomimes, or other types of creative works intended for either audiovisual reception. Where some of such works are communicated to the public, they are produced or performed or recited with the aid of performers. In such cases, there arises the interest of the performers themselves in relation to the use of their individual interpretation in the performed work.

The problem in regard to this category of intermediaries has become more acute with rapid technological developments. Where, at the very beginning of the 20th century, the performance of dramatists, actors, or musicians ended with the play or concert in which they performed, it is no longer so with the advent of the phonograph, the radio, the motion picture, the television, satellites, etc.

These technological developments made possible the fixing of performances on a variety of material, e.g. records, cassettes, tapes, films, etc. What was earlier a localized and immediate phase of a performance in a hall before a limited audience became an increasingly permanent manifestation capable of unlimited and repeated reproduction and use before an equally unlimited audience that went beyond national frontiers. The development of broadcasting and more recently, television, also had similar effects. These technological innovations have made it possible to reproduce individual performances by performing artists and to use them without their presence and without the users being obliged to reach an agreement with them.

In order to make this situation more fair for performers, the WIPO promoted the Rome Convention as a mean to incorporate new instruments in the legal systems of the countries adhering to the convention. It is defined as the right of performers to prevent fixation and direct broadcasting or communication to the public of their performance without their consent. Once a recording of the performance has been made, the performer's permission is also needed to make copies of that recording. A performer may be entitled to remuneration in respect of broadcasting and other types of communication to the public, public performance and rental of those copies.

Phonograms Producers Right

Due to the same technological changes, the development of phonograms and cassettes and, more recently, compact discs and their rapid proliferation, has forced considering the protection of producers of phonograms. In addition, there is the increasing use of records and discs by broadcasting organizations; while the use of these by the latter provides publicity for the phonograms and for their producers, these also have, in turn, become an essential ingredient of the daily programs of broadcasting organizations.

Consequently, just as the performers were seeking their own protection, the producers of phonograms began to pursue the case of their protection against unauthorized duplication of their phonograms, as also for remuneration for the use of phonograms for purposes of broadcasting or other forms of communication to the public. Their interest was formalised in the Rome Convention and is implemented by those countries that adhered to it. The producers right is defined as the right of producers of phonograms to authorize or prohibit reproduction of their phonograms and the import and distribution of unauthorized duplicates thereof.

The term "producer of phonograms" denotes a person or legal entity that first fixes the sounds of a performance or other sounds. A phonogram is any exclusively aural fixation of sounds of a performance or of other sounds. A duplicate of a phonogram is any article containing sounds taken directly or indirectly from a phonogram and which embodies all or a substantial part of the sounds fixed in that phonogram. For instance, gramophone records, magnetophone cassettes and compact discs are duplicates of a phonogram.

Broadcasters Right

Finally, there were the interests of broadcasting organizations as regards their individually composed programs. The broadcasting organizations required their own protection for these as well as against retransmission of their own programs by other similar organizations.

The Rome Convention established the broadcasters right as the right of broadcasting organizations to authorize or prohibit re-broadcasting, fixation and reproduction of their broadcasts.

Broadcasting is usually understood as meaning telecommunication of sounds and/or images by means of radio waves for reception by the public at large. A broadcast is any program transmitted by broadcasting, in other words, transmitted by any wireless means, including satellite transmissions, for public reception of sounds and of images and sounds.

Communication to the public by wire is generally understood as meaning the transmission of a work, performance, phonogram or broadcast by sounds or images through a cable network to receivers not restricted to specific individuals belonging to a private group.

Another notion, that of rebroadcasting, is either simultaneous transmission of a broadcast of a program being received from another source, or a new, deferred broadcast of a formerly recorded program transmitted or received earlier.

8.3.4 Limitations

Temporal

Copyright does not continue indefinitely. The law provides for a period of time, a duration, during which the rights of the copyright owner exist. This period begins with the creation of the work and it continues until some time after the death of the author. The purpose of this provision in the law is to enable the author's successors to have economic benefits after the author's death. It also safeguards the investments made in the production and dissemination of works.

In countries that are party to the Berne Convention, and in many other countries, the duration of copyright provided for by national law is the life of the author and not less than 50 years after the death of the author. In recent years, a tendency has emerged towards lengthening the term of protection.

In the European Union this period has been harmonised to 70 year after the death of the author. In the United States of America, in response to lobbying by major media companies, the U.S. Congress routinely extends copyright protection to works, as the copyrights are about to expire:

- **For works originally created on or after January 1, 1978**, i.e. fixed in tangible form for the first time, the copyrights is ordinarily given a term enduring for the author's life, plus an additional 70 years after the author's death. In the case of "a joint work" prepared by two or more authors that was not a "work made for hire", the term lasts for 70 years after the last surviving author's death. For works made for hire, and for anonymous and pseudonymous works, the duration of copyright will be 95 years from publication or 120 years from creation, whichever is shorter.
- **For works originally created before January 1, 1978 but not published before this date**, the duration of copyright in these works will generally be computed in the same way as in the previous case: the life-plus-70 or 95/120-year terms will apply to them as well. The law provides that in no case will the term of copyright for works in this category expire before December 31, 2002, and for works published on or before December 31, 2002, the term of copyright will not expire before December 31, 2047.
- **For works originally created and published before January 1, 1978**, the copyright endured for a first term of 28 years. During the last year of this first term, the copyright was eligible for a second renewal term of an additional 28 years. If no application was filed for renewal, the work would enter the public domain after the initial 28-year term. The current copyright law has extended the renewal term from 28 to 67 years for copyrights that existed as of January 1, 1978, making these works eligible for a total term of protection of 95 years. There is no longer a need to make the renewal filing in order to extend the original 28-year copyright term to the full 95 years. In other words, if a work was published between 1923 and 1963, the copyright owner was required to have applied for a renewal term. If they did not, the copyright expired and the work entered into the public domain. If they did apply for renewal, these works will have a 95 year copyright term and hence will enter into the public domain no sooner that 2018 (95 years from 1923). If the work was published between 1964 and 1977, there is no need to file for a renewal, and these works will automatically have a 95-year term.

Geographic

The second limitation or exception to be examined is a geographical limitation. The owner of the copyright in a work is protected by the law of a country against acts restricted by copyright done in that country. For protection against such acts done in another country, the rights holder must refer to the law of that other country. If both countries are members of one of the international conventions on copyright, the practical problems arising from this geographical limitation are very much eased.

Permitted End-User Actions

Certain end-user acts normally restricted by copyright may, in circumstances specified in the law, can be done without the authorization of the copyright owner. These exceptions to copyright should be considered as end-user privileges and not rights. However, some of them are referred to as rights, e.g. the right to quote.

Moreover, these exceptions do not mean that the exceptional usage is always free. Some of these exceptions allow use of the content without authorisation but require the user to pay compensation. For instance, in some countries, there are levies on digital recording equipment and media.

These are the main rights and usages derived from copyright exceptions:

- **Quotation Right**: the making of quotations from a protected work, provided that the source of the quotation, including the name of the author, is mentioned and that the extent of the quotation is compatible with fair practice.
- **Uses for Education**: illustration for teaching and research, uses for reproduction and communication to the public in educational institutions, libraries and archives.
- **Uses for Information Purposes**: mews incorporating other news and news incorporating other works.
- **Use for certain proceedings and ceremonies**: Administrative, judicial, etc Security proceedings Religious, official ceremonies
- **Private Copy**: the reproduction of a work exclusively for the personal and private use of the person who makes the reproduction. For instance a backup or security copy.
- **Parody and Caricature**.

- **Temporary Reproduction**: the reproduction of a work is permitted, but just temporarily, is when this copy is needed in order to carry out the technological process geared to work use. For example, when the broadcasting of a work has been authorized, many national laws permit the broadcasting organization to make a temporary recording of the work for the purposes of broadcasting, even if no specific authorization of the act of recording has been given. Another example is caching in the context of the Internet, when a work is temporarily stored in network node that is nearer to the user in order to facilitate its delivery.

In any case, these exceptions must comply with the **Bernethree-step test** set of constraints on the limitations and exceptions to exclusive rights under national copyright:

"*Members shall confine limitations and exceptions to exclusive rights to certain special cases which do not conflict with a normal exploitation of the work and do not unreasonably prejudice the legitimate interests of the rights holder*".

Non-Material Works

In some countries, works are excluded from protection if they are not fixed in some material form. Moreover, in some legal systems, the texts of laws and of decisions of courts and administrative bodies are excluded from copyright protection. It is to be noted that in some other countries such official texts are not excluded from copyright protection; the government is the owner of copyright in such works, and exercises those rights in accordance with the public interest.

Compulsory licenses

The laws of some countries permit the broadcasting of protected works without authorization, provided that fair remuneration is paid to the owner of copyright. This system, under which a right to remuneration can be substituted for the exclusive right to authorize a particular act, is frequently called a system of "compulsory licenses". Such licenses are called "compulsory" because they result from the operation of law and not from the exercise of the exclusive right of the copyright owner to authorize particular acts.

The remunerations resulting from compulsory licenses are usually collected by collective management organisations. These organisations license use of works and other subject matter that are protected by copyright and related rights whenever it is impractical for right owners to act individually. There are several international non-governmental organizations that link together national collective management organizations.

- Collective management organizations most commonly take care of the following rights:
- The right of public performance: music played or performed in discotheques, restaurants, and other public places.
- The right of broadcasting: live and recorded performances on radio and television.
- The mechanical reproduction rights in musical works: the reproduction of works in CDs, tapes, vinyl records, cassettes, mini-discs, or other forms of recordings.
- The performing rights in dramatic works: i.e. plays.
- The right of reprographic reproduction of literary and musical works: i.e. photocopying.

8.3.5 Treaties

The international protection of copyright and related rights is performed through a set of treaties managed by the WIPO. They are accessible from the WIPO's Web page, http://www.wipo.org.

There are treaties for the protection of copyright:

- Berne Convention for the Protection of Literary and Artistic Works (Berne Convention, 1886).
- WIPO Copyright Treaty (WCT, 1996).

There are also treaties for the protection of related rights:

- International Convention for the Protection of Performers, Producers of Phonograms and Broadcasting Organisations (Rome Convention, 1961)
- Convention for the Protection of Producers of Phonograms against Unauthorized Duplication of Their Phonograms (Geneva Convention, 1971)
- Convention Relating to the Distribution of Programme-Carrying Signals Transmitted by Satellite (Brussels Convention, 1974)
- WIPO Performances and Phonograms Treaty (WPPT, 1996)

Chapter 9

Conceptualisation

The objective of this chapter is to organize and structure the knowledge acquired during the specification. As it has been presented in the previous chapter, the copyright domain is a very complex one and conceptualising it is a very challenging task. In order to facilitate this, the conceptualisation process has been divided in three parts. Each part concentrates on a portion of the problem and tries to build a conceptual model for just that part.

However, each part is not independent from the rest, there are many interrelations among the parts. The conceptualisation starts from building a model for the more primitive part, the Creation Model. This model is the basis for building the conceptual models of the rest of the parts. The following step is to build the Rights Model, and then the Action Model is built on the roots of the two previous ones.

These three models constitute the conceptualisation for the fundamental part of the copyright domain. They are the basis for the contributed copyright ontology.

9.1 Creation Model

The core concepts of the ontology are those that formalise the notion of creation. Creation can be viewed from three points of view, which constitute the main points of view in almost any ontological approach. For more details see the Upper Ontologies section 2.9.1:

- *Abstract*: something that cannot exist at a particular place and time without some physical encoding or embodiment. In other ontologies it is called a mental concept.

- *Object*: it corresponds roughly to the class of ordinary objects. Object is related to the continuant or endurant concepts in some ontologies. It also includes digital objects.
- *Process*: something that happens and has temporal parts or stages. It is related to the ocurrent or perdurant concepts in some ontologies.

As we can see in Figure 9.1, these three points of view on creation can be then detailed into the different forms a creation can take. These copyright specific concepts are related through the different actions that can be performed on creations and produce new creation forms from previous ones. We will detail them when we build the action model. However, we are going to give some direct relations between creation forms together with the forms definitions. These direct relations are established by the commented actions and they just summarise the result of their application. For instance, there is the *isDerivationOf* relation between works, which results from a *Derive* action.

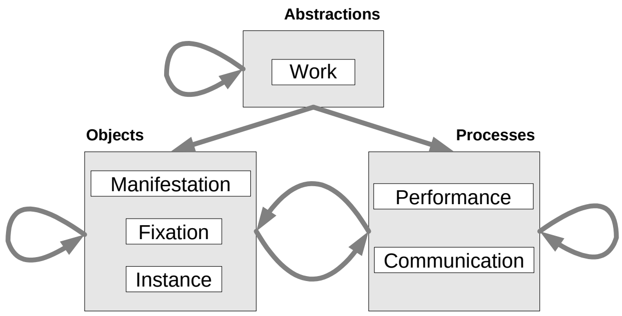

Figure 9.1: Creation model showing different views on creation

The concrete concepts in the creation model are detailed next.

9.1.1 Work

A *Work* is a distinct intellectual or artistic creation. It is original in the sense that it is not a copy of a previous work, as it is explained in the Originality section 8.3.1. Therefore, although a work is original, it might by a derivation of a previous work, i.e. it is a derived work. In other words, a work is recognised as the socially shared concept that captures the characteristics that allow the identification of the original creation among all its manifestations.

There are some internationally recognised identification schemes for works:

- **istc**: International Standard Text Code for the unique, international identification of individual textual works so that they can be uniquely distinguished from one another, regardless of the various editions, languages and/or formats in which the works appear.

- **issn**: International Standard Serial Number; ISO standard identifier for serial publications.
- **iswc**: International Standard Musical Work Code; ISO 15707 specifies a means of uniquely identifying intellectual property, such as musical and literary works. It identifies a creation, not its physical manifestations.

Work relations are detailed in Table 9.1.

Name	Domain	Range	Cardinality	Properties	Inverse
hasDerivation	Work	Work	0..N	Transitive	isDerivationOf
isDerivationOf	Work	Work	0..N	Transitive	hasDerivation

Table 9.1: *Work* relations

Examples of works are: "Mozart's The Magic Flute", "Victor Hugo's Les Misérables", etc.

9.1.2 Manifestation

A *Manifestation* is the materialisation of a work in a concrete medium, i.e. a tangible or digital object.

There are some internationally recognised identification schemes for manifestations:

- **isbn**: International Standard Book Number; ISO standard identifier for books.
- **ismn**: International Standard Music Number; ISO standard identifier for printed music.

Manifestation relations are detailed in Table 9.2.

Name	Domain	Range	Cardinality	Properties	Inverse
isManifestationOf	Manifestation	Work	1	Functional	hasManifestation
hasManifestation	Work	Manifestation	0..N	Inverse functional	isManifestationOf

Table 9.2: *Manifestation* relations

Examples of manifestations are: "the printed scores of The Magic Flute", "the editions of Les Misérables", etc.

9.1.3 Performance

A *Performance* is the expression in time of a work. Performers or technical methods might be involved in the process. In some cases there might not be any previous manifestation of the work, an *Improvisation*. Copyright just regulates the performances and improvisations done before and audience, i.e. public performances.

Performance relations are detailed in Table 9.3.

Name	Domain	Range	Cardinality	Properties	Inverse
isPerformanceOf	Performance	Manifestation	1	Functional	hasPerformance
hasPerformance	Manifestation	Performance	0..N	Inverse functional	isPerformanceOf

Table 9.3: *Performance* relations

Improvisation relations are detailed in Table 9.4.

Name	Domain	Range	Cardinality	Properties	Inverse
isImprovisationOf	Performance	Work	1	Functional	hasImprovisation
hasImprovisation	Work	Performance	0..N	Inverse functional	isImprovisationOf

Table 9.4: *Improvisation* relations

Examples of performances are: "a scenic play of The Magic Flute opera", "a screen play of any of Les Misérables films", etc.

9.1.4 Fixation

It is the materialisation of a performance in a concrete medium, a tangible or digital object.

There are some internationally recognised identification schemes for fixations:

- **isrc**: International Standard Recording Code; ISO 3901 standard identifier for audio and video recordings. It numbers each recording of a piece, not the physical item, regardless of the context or carrier on which it is issued.
- **isan**: International Standard Audiovisual Number; draft ISO standard identifier for audiovisual works.

Fixation relations are detailed in Table 9.5.

Name	Domain	Range	Cardinality	Properties	Inverse
isFixationOf	Fixation	Performance	1	Functional	hasFixation
hasFixation	Performance	Fixation	0..N	Inverse functional	isFixationOf

Table 9.5: *Fixation* relations

Examples of fixations are: "a sound recording of The Magic Flute opera", "any motion picture of Les Misérables", etc.

9.1.5 Instance

It is the reproduction, or copy, of a manifestation, a fixation or another instance, an object.

There are some internationally recognised identification schemes for instances:

- **ean13** : European Article Number.
- **upc**: Universal Product Code.

Instance relations are detailed in Table 9.6.

Name	Domain	Range	Cardinality	Properties	Inverse
isInstanceOf	Instance	Manifestation or Fixation	1	Functional	hasInstance
hasInstance	Manifestation or Fixation	Instance	0..N	Inverse functional	isInstanceOf

Table 9.6: *Instance* relations

Examples of instances are: "a CD of The Magic Flute", "a DVD of Les Misérables", etc.

9.1.6 Communication

A *Communication* is the transmission of a work among places at a given time. It is a process performed when the public is not present at the place and or time where the *Communication* originates. It includes broadcasts, i.e. one to many, but also communications from a place and at a time individually chosen. The *Communication* can be the transmission of a previous *Fixation*, i.e. a *RecordedCommunication*, or the live transmission of a *Performance*, i.e. a *LiveCommunication*.

Name	Domain	Range	Cardinality	Properties	Inverse
isCommunicationOf	RecordedCommunication	Fixation	1	Functional	hasCommunication
hasCommunication	Fixation	RecordedCommunication	0..N	Inverse functional	isCommunicationOf

Table 9.7: *RecordedCommunication* relations

The relations that are specific to *RecordedCommunication* are detailed in Table 9.7.

Those specific to *LiveCommunication* are detailed in Table 9.8.

Name	Domain	Range	Cardinality	Properties	Inverse
isCommunicationOf	LifeCommunication	Performance	1	Functional	hasRetransmission
hasCommunication	Performance	LifeCommunication	0..N	Inverse functional	isRetransmissionOf

Table 9.8: *LiveCommunication* relations

Examples of instances are: "a radio broadcast of a sound recording of The Magic Flute", "an internet streaming of the film Les Misérables", etc.

9.1.7 Example

For instance, if we consider the creation "Les Misérables", we can observe it from these three perspectives taking different forms. From the *Object* view, we can see the original manuscript by Victor Hugo as a *Manifestation*; there are other manifestations of posterior adaptations, like a script for a film or theatre representation. Then, there is the *Fixation* of the film and *Instances* like a DVD copy of the film fixation or a book reproducing a manifestation. From the *Process* perspective, the theatre representation or the film projection in a cinema are *Performances*. Its broadcasting is a *Communication*. All the previous have in common what is socially identified as the Victor Hugo's *Work*. This is from the abstract perspective and it represents what we grasp as common in the different manifestations, performances, fixations and instances, i.e. what allows us saying that they are from the same *Work*.

9.1.8 Other Creation Models

INDECS

The creation types that compose the INDECS creation model [100]. Their interrelations are detailed in Figure 9.2:

- **Abstraction**: A creation that is a concept; an abstract creation whose existence and nature are inferred from one or more expressions or manifestations.
- **Manifestation**: An artefact containing an infixion of an expression.
- **Expression**: An event that is a creation.
- **Artefact**: A creation that is a thing.
- **Item**: A single instance of an artefact.

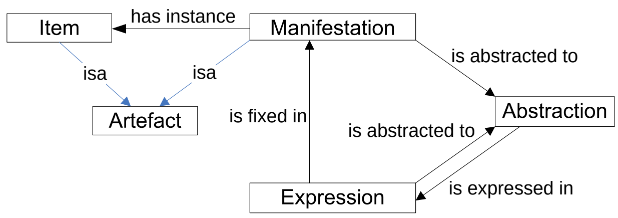

Figure 9.2: INDECS Creation Model

MPEG-21 Vision

In the MPEG-21 document vision document [65], the creation model is defined by:

- **Abstraction**: distinct intellectual or artistic creation or concept.
- **Expression**: Intellectual or artistic realisation of an Abstraction.
- **Manifestation**: The physical or digital embodiment of an Expression.
- **Asset**: Manifestation, i.e. a physical or digital embodiment of an Expression.
- **Resource**: Individually identifiable Asset such as a video or audio clip, an image, or a textual Asset.

MPEG-21 RDD

The RDD specification [67] defines the following terms, and it is shown in Figure 9.3:

- **Abstraction**: A Conceptual Resource Derived from a Manifestation.
- **Expression**: An Event in which a Resource is expressed.
- **Manifestation**: A Perceivable Resource.
- **Resource**: An Entity involved in a Context, other than as an Agent, Time or Place.

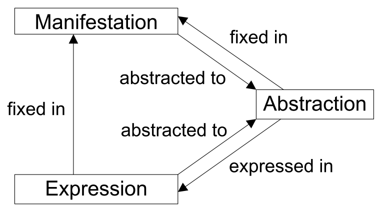

Figure 9.3: MPEG-21 RDD Creation Model

9.2 Rights Model

From the legal point of view, the WIPO recommendations have been followed and the copyright notions it defines at the international level have been incorporated into our ontological framework. Table shows the included rights hierarchy starting from Copyright. There are the economic rights plus the moral rights, as promoted by the WIPO, and the copyright related rights.

The more important rights in the DRM context are the economic rights as they are related to productive and commercial aspects of copyright. Each of these rights regulates an abstract set of actions:

- **Reproduction Right**: regulates actions that produce replicas of a given object, i.e. *Instances*. Examples of reproduction are the mass production of CD copies from an audio recording master, to scan a book in order to produce a digitalisation of it or to download a digital file into the local hard disk.
- **Distribution Right**: regulates actions geared to distribute previously made copies incorporated in tangible articles. The ownership of the corresponding physical support can be trans-

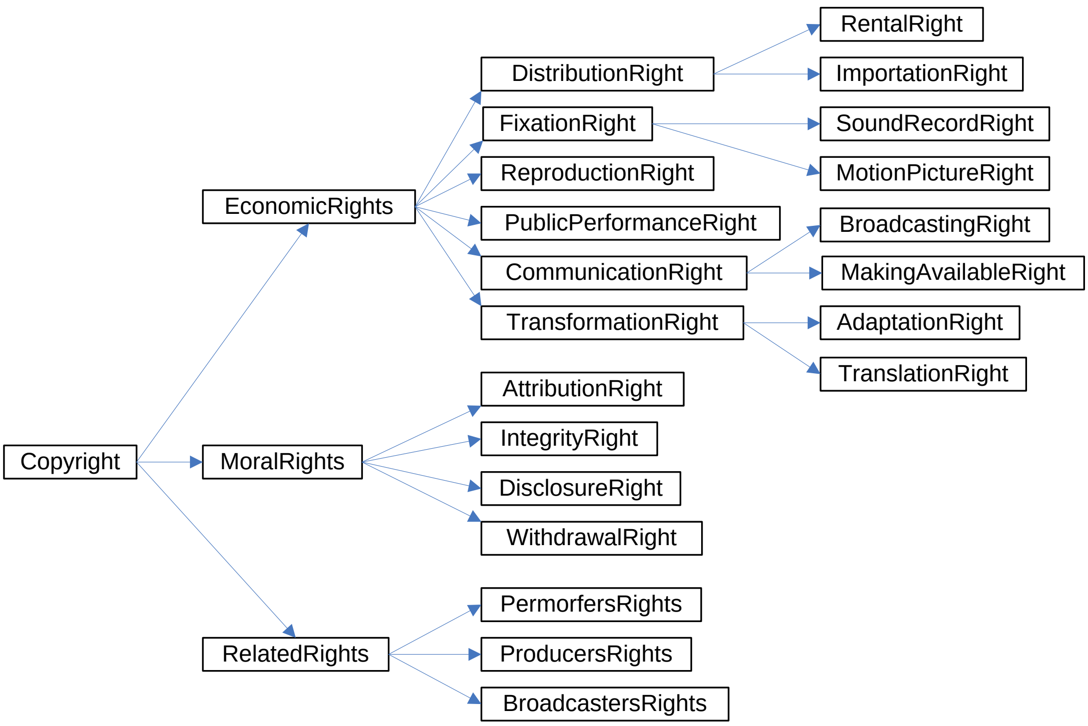

Figure 9.4: Copyright hierarchy

ferred permanently, i.e. the distribute act is a sale, or just temporally, i.e. a rent if there is a significant economic compensation or a loan if not.

- **Public Performance Right**: regulates *Performances* of works when they are made in public, i.e. before an audience.
- **Fixation Right**: regulates the materialisation of a *Performance* into an object that constitutes a *Fixation*. Common fixations are motion pictures and sound recordings, which are governed by the corresponding **Sound Recording Right** and **Motion Picture Right**.
- **Communication Right**: regulates the realisation of C*ommunications* of works, including wire or wireless means and those realised from a place and at a time individually selected. This right is concretised into **Broadcasting Right**, when the communication is massive, and **Make Available Right**, when the communication is individually chosen.
- **Transformation Right**: regulates actions that generate new works from previously existing ones. The results of this kind of actions are considered new works, and not mere reproductions, because they contribute something new, i.e. they are original as detailed in the Originality section 8.3.1. This right is concretised into the **Adaptation Right** and the **Translation Right**. The former creates a new work of a different type than the original one, e.g. a film from a novel. The latter generates a work of the same type but in a different language.

These rights are conceded to the author or promoter of the creation by the mere action of bringing the work into existence. From this initial situation, it is possible to transfer, or at least license, the economic rights to third parties. This is combined with the act of consumption of end-users and motivates value chains to arise.

On the other hand, moral rights are always held by the creator and cannot be commercially exploited. They are not present in all legal systems. However, WIPO treaties are promoting some of them in order to improve worldwide copyright law harmonisation:

- **Attribution Right**: the right to claim authorship of the work.
- **Integrity Right**: the right to object to any distortion, mutilation or other modification of, or other derogatory action in relation to, the work which would be prejudicial to the author's honour or reputation.
- **Disclosure Right**: exclusive right to disclose the work.
- **Withdrawal Right**: exclusive right to withdraw the work. In order to show the capabilities of the ontological framework, a complete withdrawal scenario is modelled in the Withdrawal Scenario section 10.5.2.

Finally, there are the rights of other persons also involved in the exploitation of works. Performers, producers and broadcasters make a significant contribution in order to make works reach end-users. Their contribution is also protected by some rights related to copyright, the **Related Rights** or **Neighbouring Rights**:

- **Performers Rights**: performers have exclusive Fixation, Communication, Reproduction, Public Performance and Distribution Rights over their performances. These rights, when the performance is of a copyrighted work, will be in addition to the rights of copyright owners with respect to the performance and subsequent exploitation of the performance.
- **Producers Rights**: producers have exclusive Reproduction Right over their fixations and exclusive Distribution Right over the resulting copies. As before, these rights will be in addition to the rights of copyright owners.
- **Broadcasters Rights**: broadcasters have exclusive Broadcasting, Fixation and Reproduction Rights over their broadcasts. Here broadcasting of a broadcast is understood as re-broadcasting. Re-broadcasting is either simultaneous transmission of a broadcast of a program being received from another source or a new deferred broadcast of a formerly recorded program transmitted or received earlier. These rights must be considered in addition to the rights of copyright owners when applicable.

End-users do not hold any right. They just consume creations, i.e. they use them, and uses are not covered by copyright. However, this does not mean that end users can do whatever they want, they should not realise actions that require copyright. Moreover, they might be subject to special conditions under which they have acquired the permission to use a creation, e.g. a film that can only be viewed a fixed number of times and thus is cheaper than a DVD reproduction. This kind of conditions is not regulated by copyright, it is established by the usage agreements among end-users and content providers. This kind of agreements are the kind of expressions captured by common rights expression languages as it is explained in the DRM and the Law section **??**. On the other hand, the contributed copyright ontology is general enough to be able to model copyright related actions and also end-users actions like those established in content provision agreements.

However, there are some aspects of end-users activity that are regulated by copyright. End-users have some special permissions that grant them the possibility to perform some actions otherwise forbidden by copyright, although this does not mean that the user must pay a compensation if they are exercises, e.g. levies on digital recording equipment and media. These exceptions to copyright should be considered as end-user privileges and not rights. However, some of them are referred to as rights, e.g. the right to quote. Moreover, they are modelled as rights in this conceptualisation in order to build a more homogeneous model as it is explained in the Action Model section 9.3. The end-users rights are:

- **Quotation Right**: the making of quotations from a protected work, provided that the source is mentioned and that the extent of the quotation is compatible with fair practice.
- **Education Right**: illustration for teaching and research, uses for reproduction and communication to the public in educational institutions, libraries and archives.
- **Information Right**: news incorporating other news and news incorporating other works.
- **Official Act Right**: use for certain administrative, judicial or security proceedings and religious or official ceremonies.
- **Private Copy Right**: the reproduction of a work exclusively for the personal and private use of the person who makes the reproduction, e.g. a backup.
- **Parody Right**: use for parody and caricature.
- **Temporary Reproduction Right**: ephemeral reproductions required for facilitating some technological processes geared towards work usage, e.g. internet caches.

9.3 Action Model

As it has been already shown, the ontological framework considers creation in its different forms and copyright rights. Trying to go to the more primitive elements in this framework, we can see that rights define actions packages that they regulate. Moreover, the different forms a creation can take are organised in a creation life cycle that is performed by these same actions, at least in the part that is governed by copyright. Figure 9.5 situates these actions in the creation life cycle.

First of all, two actions that take original creations into existence can be identified:

- *manifest*: this action generates a manifestation from a work.
- *improvise*: this action directly generates a performance from a work, without a previous manifestation.

These actions are generalisations of the kinds of actions governed by the different copyright economic rights:

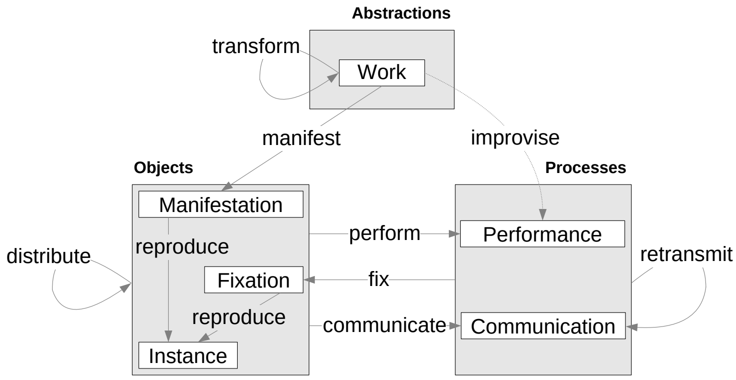

Figure 9.5: Actions in the creation life cycle

- Reproduction Right: *reproduce*, commonly speaking *copy*.
- Distribution Right: *distribute*. More specifically *sell*, *rent* and *lend*.
- Public Performance Right: *perform*; it is regulated by copyright when it is a public performance and not a private one.
- Fixation Right: *fix*, or *record*.
- Communication Right: *communicate* when the subject is an object or *retransmit* when communicating a performance or previous communication, e.g. a re-broadcast. Other related actions, which depend on the intended audience, are *broadcast* or *make available*.
- Transformation Right: *derive*. Some concretisations are *adapt* or *translate*.

There are also generic actions for the other copyright rights. For the related rights the actions are the same than for the economic rights, as the rights of performers, producers and broadcasters are also economic rights limited to their contributions. For the moral rights there are:

- Attribution Right: *attribute*.
- Integrity Right: *oppose to change*.
- Disclosure Right: *disclose*.
- Withdrawal Right: *withdraw*.

Finally there are the actions performed by end-users that are subject to copyright exceptions and have been modelled as rights. All these actions are specific kinds of the actions governed by the economic rights that exceptionally are not subject to them:

- Quotation Right: *quote*, a limited extent reproduce action of a source protected work, which is clearly mentioned.
- Education Right: *educational act*, any reproduce, communicate or perform action with educational or research purposes.
- Information Right: *inform*, any copyright governed act with informative purposes.
- Official Act Right: *official act*, any copyright governed act that is part of an official act.
- Private Copy Right: *reproduce privately*, a reproduce act that produces a reproduction solely for private consumption.
- Parody Right: *parody*, any copyright governed act with parody or caricature purposes.
- Temporary Reproduction Right: *reproduce temporally*, a reproduce act that produces a temporal reproduction.

From this life cycle many value chains can be built but, in order to do that, we must also consider the *Transfer* and *Use* actions. The former is the basic action to model the flux of rights through the value chain, even if it is a real transfer or a temporary one, i.e. a license. The latter models any kind of consumption of a creation in one of its object or process forms.

The end-user consumption actions can be referred to generically using the generic *use* action. However, some specialisations are given below depending on the kind of creation form they consume. It can be seen that creations that are objects are generically bought (or rented or lent) and those that are processes are generically accessed:

- A manifestation: to *buy* a picture or a sculpture.
- An instance: to *buy* a book, CD or DVD.
- A performance: to *assist* to a cinema projection, orchestra recital or an art exhibition.
- A communication: generically to *access* a communication. More concretely:
  - A broadcast: to *tune* a TV channel or a radio station.
  - Something made available: to *access* a web page or an internet on-demand audio or video stream.

Finally, in order to complete the action model, we have also included negotiation actions: *Offer*, *Agree* and *Counteroffer*. These actions are necessary to construct the value chains. They can model the negotiation processes necessary in order to achieve the agreements that finally establish the value-chain shape. The agreements include transfers of rights but also usage licenses for actions governed by copyrights or not, i.e. end-user creations consumption actions.

9.3.1 Value Chain Example

The previously introduced pool of primitive actions can be combined in order to build different value chains in the copyright domain. Figure 9.6 shows how we can build a model for the value chain of serials adapted from literary works. The ovals represent the different roles involved, which perform the actions they are linked to.

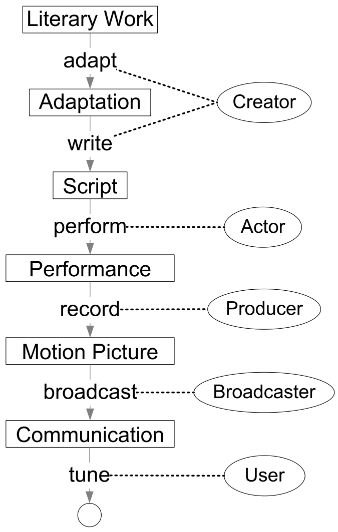

Figure 9.6: Serials adapted from literary works value chain

First of all, the creator adapts the original literary work, e.g. Alexandre Dumas' "The Count of Monte Cristo", in order to produce a serial. The resulting adaptation is realised as a script that is performed by some actors, e.g. Gerard Depardieu, and recorded into a motion picture. This motion picture is finally broadcasted to users who can tune the resulting communication. This is just the skeleton of the value chain. In order to give a more detailed model each step in the value chain can be modelled as an event for the corresponding action.

9.3.2 Case Roles

Actions are not isolated entities, they are related to a bunch of entities that take part or are affected by the action. Moreover, there are space-time coordinates that situate actions. In order to capture all these details and facilitate their modelling, they are modelled as verbs and the ontology incorporates concepts from the linguistics field related to the classification of verbs and how they are related to other linguistic components.

These relations are called thematic roles or case roles [106, 29] and are generically classified into initiator, resource, goal and essence. In Figure 9.7, it is shown at the top the generic case roles and at right the kinds of verbs they are related to. These kinds of verbs define verbs facets, not disjoint classes of verbs, and concretise the general thematic roles as shown in each row. Therefore, the same verb can present one or more of these facets. For instance, the play verb can show the action, temporal and spatial facets in a particular sentence.

	initiator	**resource**	**goal**	**essence**
Action	agent, effector	instrument	result, recipient	patient, theme
Process	agent, origin	matter	result, recipient	patient, theme
Transfer	agent, origin	instrument, medium	experiencer, recipient	theme
Spatial	origin	path	destination	location
Temporal	start	duration	completion	pointInTime
Ambient	reason	manner	aim, consequence	condition

Figure 9.7: Case roles

The general case roles are:

- **Initiator**: a participant that determines the direction of the process from the beginning (Determinant/Source).
- **Resource**: a participant that must be present at the beginning of the process, but not necessarily through it, and does not actively control what happens (Immanent/Source).
- **Goal**: a participant that determines the direction of the process from the end (Determinant/Product).
- **Essence**: a participant that must be present at the end of the process, but not necessarily through it, and does not actively control what happens (Immanent/Product).

There are 23 specific case roles:

- **Agent**: the voluntary initiator of an event.
  Example: *[Eve] bit an apple*.
- **Effector**: the involuntary initiator of an event.
  Example: *[The tree] produced new leaves*.
- **Instrument**: a resource that is not changed by an event.
  Example: *[The key] opened the door*.
- **Result**: an inanimate goal of an event.
  Example: *Eric built a [house]*.
- **Recipient**: an animate goal of an event.
  Example: *Sue sent the gift to [Bob]*.
- **Patient**: an essential participant that undergoes some structural change as a result of the event.
  Example: *The cat swallowed [the canary]*.
- **Theme**: an essential participant that may be moved, said, or experienced, but is not structurally changed.
  Example: *Billy likes [the Beer]*.
- **Origin**: a passive determinant source.
  Example: *The chapter begins on [page 20]*.
- **Matter**: a resource that is changed by the event.
  Example: *The gun was carved out of [soap]*.
- **Medium**: a physical resource for transmitting information, such as the sound of speech or the electromagnetic signals that transmit data.
  Example: *Bill told Boris by [phone]*.
- **Experiencer**: an active animate goal of an experience.
  Example: *[Yojo] sees the fish*.
- **Path**: a spatial resource.
  Example: *The pizza was shipped via [Albany and Buffalo]*.
- **Destination**: a goal of a spatial process.
  Example: *Bob went to [Danbury]*.

- **Location**: an spatial essential participant.
  Example: *Vehicles arrive at [a station].*
- **Start**: a determinant temporal source.
  Example: *Bill waited from [noon] to three.*
- **Duration**: a resource of a temporal process.
  Example: *The truck was serviced for [5 hours].*
- **Completion**: the goal of a temporal process.
  Example: *Mary waited until [noon].*
- **PointInTime**: a temporal essential participant.
  Example: *[At 5:25 PM], Erin left.*
- **Reason**: the initial cause of an event.
  Example: *The airplane fell due to [a malfunction].*
- **Manner**: the way an event develops.
  Example: *The car moved [slowly].*
- **Aim**: the voluntary goal of an action.
  Example: *The player tries [to hit the ball].*
- **Consequence**: the involuntary goal of an action.
  Example: *The player missed [to hit the ball].*
- **Condition**: the necessary circumstance for an action.
  Example: *In order to see the film [you must pay the entrance].*

Figure 9.8 shows an example of an action modelled as a verb. In this case it is a creation action where a manifestations is realised. At the centre there is the box representing the verb. Its type is defined in bold, it is a *Manifest*. Case roles relate the action verb to its participants and context, they are shown as arrows from the verb to the participants. Participants are also represented by boxes. The *agent* participant is represented by a box containing its identifier. For *theme* participant, it is specified its type, it is a *Work*, and its identifier. The same is done for the *result* participant. In this case it is a *Manifestation* and a universal identifier for manifestations is used. Finally, the verb is contextualised in time by a *pointInTime* participant that specifies the concrete time at which the action took place.

As it can be noted, types are shown in bold and resource identifiers in normal text. The type is only specified when it is relevant in the context of the copyright ontology. On the other hand, identifiers are usually provided. However, there are unidentified resources when an identifier is not required, i.e. anonymous resources, or a local one might be used, i.e. there is not universal identification scheme at hand.

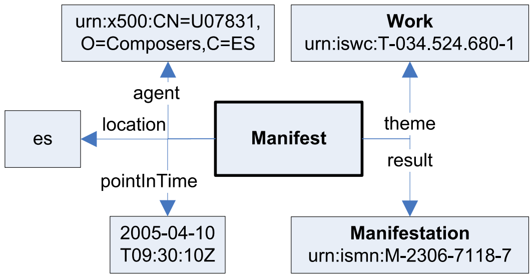

Figure 9.8: Verb modelling example using case roles for a *Manifest* action

This kind of verb models based on the concepts and relations from the copyright ontology constitute the basic building blocks of the contributed "Rights Expression Language". The next sections give more detail about each one of the previous actions and how they can be modelled as verbs using the case roles and other concepts in the ontology. Moreover, the are axioms and rules that capture the dynamic knowledge associated to the actions. It is also important to note that, in order to perform these actions, the involved agents must hold the necessary rights or licenses. In the Implementation chapter 10 it is shown how the copyright ontology facilitates checking this.

9.3.3 Rights-Generation Actions

Manifest

In the previous Figure 9.8 there was an example of manifest action, i.e. when the author produces a manifestation of a work. Table 9.9 details the involved case roles for the *Manifest* verb.

There are one or more agents. In the case of employed authors and works made for hire, the employer, and not the employee, is considered the author of the work and thus becomes the rights holder. That is why the range of *agent* is a person, natural or legal. Therefore, *agent* is the person considered to be the author of the work from the legal point of view. Each manifest event has, from the legal point of view, just one *theme*, one *result* and one *pointInTime*. It is also necessary to state a *location*, which will determine the legal system that will govern copyright.

If this action brings into existence the first manifestation of a work, the manifest event implies that the creator becomes the holder of all the copyright rights on the creation. Multiple creators are also allowed in order to cope with works realised in collaboration as specified in the Kinds of

Case role	**Range**	**Cardinality**
agent	Person (Natural or Legal)	1..N
theme	Work	1
result	Manifestation	1
pointInTime	e.g. ISO8601	1
location	e.g. ISO3166, Country, etc.	1
...	...	...

Table 9.9: *Manifest* case roles

Works section 8.3.1. All this is modelled by Manifest-Rule in Table 9.10. In order to check that this is the first manifestation of the work, the rule tests if there is not a statement asserting that there is a manifestation of the work yet. All rules are written down using a common first order logic notation, which is described in the Knowledge Representation chapter 4.6.

(∀v:Manifest) ((∃mr:MoralRights)(∃er:EconomicRights) ((∀p:Person)(∀m′:Manifestation)(∀w:Work)(∀m:Manifestation)(∀t:Time) (¬isManifestationOf(m′,w)∧agent(v,p)∧theme(v,w)∧result(v,m)∧ pointInTime(v,t)∧location(v,l) → agent(mr,p)∧agent(er,p)∧essence(mr,w)∧essence(er,m)∧ start(mr,t)∧start(er,t)∧location(mr,l)∧location(er,l)∧isManifestationOf(m,w)))

Table 9.10: Manifest-Rule: assign author rights

Figure 9.9 shows the situation resulting from applying this rule due to the manifest action in Figure 9.8.

Rights

As it can be noted from the previous Figure 9.9, *MoralRights* and *EconomicRights* are also modelled as verbs, with an *agent*, *start* and *essence* case roles. This is so because, as it has been said in the Rights Model section 9.2, rights are modelled as a package of governed actions and they are hierarchically structured. In order to make that a given right, hold by a given party, on a given creation, from a given date, etc. packages the concrete actions that this concrete right governs, it is also modelled as a verb. As it will be shown in the Implementation chapter 10, this modelling decision would facilitate to check if a given concrete action is granted by a concrete right. This check would be reduced to prove that the rights subsumes, i.e. "packages" or includes, the concrete action.

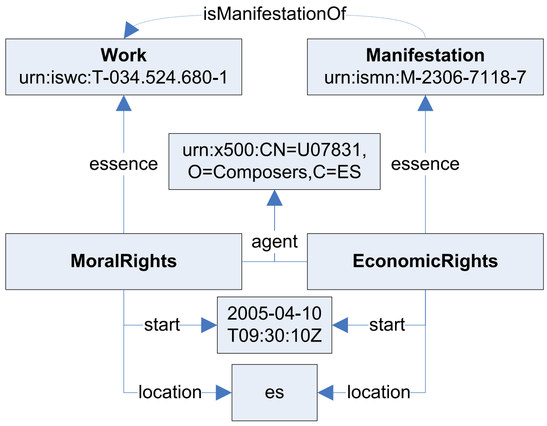

Figure 9.9: Rights situation resulting from the manifestation of a work

Another consequence of this approach is that the generic *essence* case role is used to relate the right to the *Manifestation* or *Work*. The generic case role is used in order to make the right situation expression subsume concrete actions that are granted by the right in which the *essence* of the right can appear as the *theme*, i.e. the essential unchanged participant of the action, or the *patient*, i.e. the essential changed participant of the action. The case roles for *MoralRights* are detailed in Table 9.11, and those for *EconomicRights* in Table 9.12.

Case role	**Range**	**Cardinality**
agent	Person	1..N
essence	Work	1
start	e.g. ISO8601	1
duration	e.g. author life + 70 years	1
location	e.g. ISO3166, Country, etc.	1
...	...	...

Table 9.11: *MoralRights* case roles

For more concrete rights, the *essence* is the corresponding creation form as specified in the Rights Model section 9.2.

Improvise

This action directly produces a *Performance* from a *Work* without a previous *Manifestation*. As it has been shown in the specification, not all legal systems consider that the resulting performance is

Case role	Range	Cardinality
agent	Person	1..N
essence	Manifestation	1
start	e.g. ISO8601	1
duration	e.g. author life + 70 years	1
location	e.g. ISO3166, Country, etc.	1
...	...	...

Table 9.12: *EconomicRights* case roles

subject to copyright. Moreover, there is the additional difficulty to demonstrate authorship if there is not material evidence. Despite these limitations, the conceptual model captures improvisations. Table 9.13 shows the case roles that are associated to the *Improvise* verb.

Case role	Range	Cardinality
agent	Person (Natural or Legal)	1..N
theme	Work	1
result	Performance	1
pointInTime	e.g. ISO8601	1
location	e.g. ISO3166, URL, etc.	1
...	...	...

Table 9.13: *Improvise* case roles

Therefore, taking into account the related case roles, it is possible to model *Improvise* as shown in Figure 9.10. The required rule to associate the rights to the author of an improvisation is like the Manifest-Rule in Table 9.10 but it look for *Improvise* instances instead of *Manifest* ones.

Another consequence of improvisations is that there is no universal identifier for performances and that it is very difficult to get a work identifier for the underlying work, e.g. a ISWC. For instance, in order to get a ISWC for a musical work, it is necessary to provide the score and the lyrics of the song if applicable, which are the previous manifestations of the song performance.

Derive

This correspond to the act changing a work protected by copyright in order to generate a new work. The resulting derivations are themselves works protected by copyright. *Derive* is thus also a rights

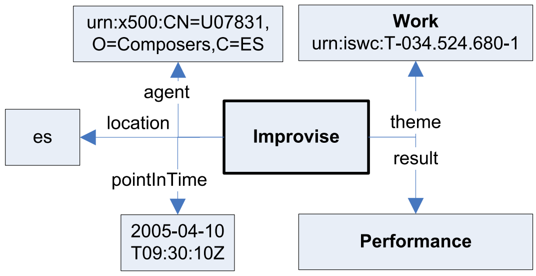

Figure 9.10: Model for an *Improvise* action

generation action, i.e. it causes new rights to arise as a new *Work* is generated and materialised in its first *Manifestation*, which *results* from the *Derive* action.

The action is required to produce the first *Manifestation* of the *Work* in order to have legal implications. If there is not a *Manifestation*, the derivation takes part just in the field of ideas, which are not regulated by copyright as it has been explained in the Specification chapter 8. The new *Work* is pointed by an *aim* case role and the *Manifestation* by a *result* case role. Table 9.14 shows all them for the *Derive* verb.

Case role	**Range**	**Cardinality**
agent	Person (Natural or Legal)	1..N
theme	Work	1
aim	Work	1
result	Manifestation	1
pointInTime	e.g. ISO8601	1
location	e.g. ISO3166, URL, etc.	1
...	...	...

Table 9.14: *Derive* case roles

The *Derive* action is then quite similar to *Manifest*. However, there are two *Works* involved. The one that is the *theme* of the verb was there before the *Derive* and serves as the source of the transformation. The *aim* one is new an it is derived from the previous one and this should be marked in the rights situation arising from the derivation. It is modelled by Manifest-Rule in Table 9.15.

The previous rule generates a chain of relations that models the corresponding chain of derivations. It is important to keep this chain because from the legal point of view it is required, prior to

(∀v:Derive) ((∃mr:MoralRights)(∃er:EconomicRights) ((∀p:Person)(∀w:Work)(∀w':Work)(∀m:Manifestation)(∀t:Time) (w≠;w')∧agent(v,p)∧theme(v,w)∧result(v,m)∧aim(v,w')∧pointInTime(v,t)∧location(v,l) → agent(mr,p)∧agent(er,p)∧essence(mr,w')∧essence(er,m)∧start(mr,t)∧start(er,t)∧ location(mr,l)∧location(er,l)∧isManifestationOf(m,w')∧isDerivationOf(w',w)))

Table 9.15: Derive-Rule: assign author rights

any derivation, that the author of the derivation follows this chain in order to get the authorisations from the owner of the copyright of the original work through all derived works until the current derivation.

There are two common kinds of derivation of copyrighted works, *adapt* and *translate*.

Translate

To *Translate* means to express a work in a language other than that of the original version. Therefore, it is possible to just chain the language characteristic from the original *Work* to the new one.

Adapt

To *Adapt* is generally understood as the modification of a *Work* from one type of *Work* to another, for example adapting a novel so as to make a motion picture, or the modification of a *Work* so as to make it suitable for different conditions of exploitation, for example adapting an instructional textbook originally prepared for higher education into an instructional textbook intended for students at a lower level. Therefore, *Adapt* is a more general kind of actions that *Translate* as many content characteristics can change.

9.3.4 Economic Rights Actions

Perform

This corresponds to the action of public performance. To perform a work means to recite, render, play, dance, or act it, either directly or by means of any device or process or, in the case of a motion picture or other audiovisual work, to show its images in any sequence or to make the sounds accompanying it audible.

It is important to note that the performance is considered public when it takes place at a place open to the public or at any place where a substantial number of persons outside of a normal circle of a family and its social acquaintances is involved. Therefore, it will not be considered public when it is performed in a strictly domestic domain.

Due to this requirement, the case roles for the *perform* verb include the restriction of the kind of location to *PublicPlace*, as it is shown in Table 9.16. Public places are a kind of places as understood in the Copyright Ontology, i.e. they include physical locations but also Internet locations like URLs. The special characteristic of this places is that they are accessible to the public.

Case role	**Range**	**Cardinality**
agent	Person (Natural or Legal)	1..N
theme	Manifestation	1
result	Performance	1
pointInTime	e.g. ISO8601	1
location	PublicPlace	1
...	...	...

Table 9.16: *Perform* case roles

On the other hand, the *agent* case roles continues to be restricted to *Person*, *Natural* or *Legal*. This is due to the fact that a musician playing an instrument may be the *agent* of a *Perform*, but a film exhibition company that plays a film in one of its theatres is also the *agent* of a *Perform*. Concrete *Perform* events can be then modelled, as it is shown in Figure 9.11.

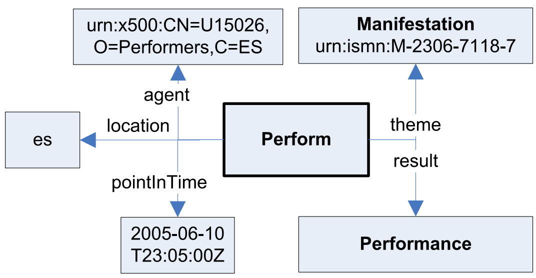

Figure 9.11: Model for a *Perform* action

Record

Generically, a work is considered fixed when it is stored on some medium in which it can be perceived, reproduced, or otherwise communicated. For example, a song is considered fixed when it is written down on paper. The paper is the medium on which the song can be perceived, reproduced and communicated.

It is not necessary that the medium be such that a human can perceive the work, as long as the work can be perceived by a machine. Thus, the song is also fixed the moment the author records it onto a cassette tape. Similarly, a computer program is fixed when stored on a computer hard drive.

From the legal point of view, the fixations that are governed are those that store performances of works. The more important kinds of fixations are sound recordings and motion pictures. Consequently, the *theme* of a *Record* is a *Performance* as it is shown in Table 9.17. However, it is not necessarily a public performance. For instance, it can be a studio performance, which is governed by copyright because this fixation is intended for commercial exploitation. Finally, it is necessary to assert that the *FixationisFixationOf* the *Performance*.

Case role	**Range**	**Cardinality**
agent	Person (Natural or Legal)	1..N
theme	Performance	1
result	Fixation	1
pointInTime	e.g. ISO8601	1
location	e.g. ISO3166, URL, etc.	1
...	...	...

Table 9.17: *Record* case roles

Record actions can be modelled using theses building blocks as shown in Figure 9.12.

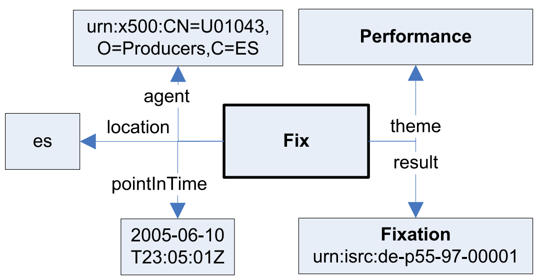

Figure 9.12: Model for a *Record* action

Copy

This action is formally known as *Reproduce*. However, it is commonly referred to as *Copy* and this term is the one that is going to be used in the ontology in order to improve its usability. Copies have been traditionally the basic medium for *Work* commercialisation. They are produced from a *Manifestation,* from a *Fixation* of a *Performance* or from another *Instance*. Therefore, these are the *theme* of the *Copy* verb as it is shown in Table 9.18.

The result is an *Instance* that is the item employed for the physical commercialisation of works, i.e. when a physical item is used as the vehicle to make the *Work* arrive to its consumers. For example, the making of copies of a protected work is the act performed by a publisher who wishes to distribute copies of a text-based work to the public, whether in the form of printed copies or digital media such as CD-ROMs.

Case role	**Range**	**Cardinality**
agent	Person (Natural or Legal)	1..N
theme	Manifestation OR Fixation OR Instance	1
result	Instance	1
pointInTime	e.g. ISO8601	1
location	e.g. ISO3166, URL, etc.	1
...	...	...

Table 9.18: *Copy* case roles

Copy actions are modelled using these restrictions on the related case roles as shown in Figure 9.13.

Figure 9.13: Model for a *Copy* action

Distribute

Once *Instances* of a *Work* have been produced, they are distributed to the public. Therefore, there is a *recipient* of the *Instance*, which is the *theme* of the *Distribute* action, see Table 9.19. There is also the *duration* case roles, which is used to model the duration of the transfer of ownership that is implicit in the *Distribute* action if it is not permanent.

Distribute actions can be modelled using theses building blocks as shown in Figure 9.14.

Case role	**Range**	**Cardinality**
agent	Person (Natural or Legal)	1..N
theme	Instance	1
recipient	Person (Natural or Legal)	1..N
pointInTime	e.g. ISO8601	1
location	e.g. ISO3166, URL, etc.	1
duration	e.g. ISO8601	0..1
...	...	...

Table 9.19: *Distribute* case roles

Figure 9.14: Model for a *Distribute* action

Moreover, *Distribute* is concretised into *Sell*, *Rent* and *Lend*. In the first case, the ownership of the corresponding physical support is transferred permanently, i.e. the distribute action is a sale. Therefore, there is not a *duration* case role associated to the *Distribute* verb.

As it has been explained in the Distribution Right section 8.3.2, the right of distribution is usually subject to exhaustion upon first sale or other transfer of ownership of a particular copy. This means that, there are are not *Distribute* actions that have a previously distributed *Instance* as theme. This is captured by the Distribute-Constraint in Table 9.20.

(∀v:Sell)(∀i:Instance)(theme(v,i) → ¬(∃v′:Sell)(theme(v′,i)∧v≠;v′))

Table 9.20: Distribute-Constraint: first sale exhaustion

If the ownership transfer is temporal, the *Distribute* action is concretised into *Rent*, if there is a significant economic compensation, or *Lend*, if the compensation is not significant. The *duration* case roles specifies the temporal span of the ownership transfer. The compensation is associated to the corresponding verb using and additional case role called *condition*. This case role introduces something that must be satisfied in order to enable the main verb. On the other hand, the distinction between a *Rent* and a *Lend* must be determined at modelling time because to determine if a compensation should be considered significant or not lies outside of the scope of this work.

Figure 9.15 shows an example of a *Rent* action with an associated *Transfer* action that serves as a compensation. In this case, it is an economic compensation because it specifies a *Transfer* of an amount money in a specified currency, the *theme*, between the *recipient* and the *agent* of the original *Distribute* action.

Figure 9.15: Model for a *Rent* action

Some copyright laws include a right to control importation of copies as a means of preventing erosion of the principle of territoriality of copyright. In order to model this kind of controls, the *origin* case role can be specified in order to clarify the source territory of the *Instance*. This information can be deduced from the *location* where the *Copy* took place. Therefore, an importation control for distribution can be easily formalised.

Communicate

This verb includes any communication to the public of the originals or copies of works, including wire or wireless means and "the making available to the public of works in a way that the members of the public may access the work from a place and at a time individually chosen by them". The quoted expression covers, in particular, on-demand interactive communication through the Internet. On the contrary to *Perform,* this action just covers all communication to the public not present at the place where the communication originates.

The required case roles in order to model this verb include *result*, which relates the verb to the resulting *Communication*, and the *medium* case role, which relates the communication medium to the verb. The communication medium can be used as a mean to determine the intended recipients of the communication, e.g. if the communication medium is the World Wide Web, the intended audience is the whole WWW. This is true even if there are some access control mechanism as long as the may be open for any WWW user after fulfilling the access requirements, e.g. a pre-payment is required. Another more specific communication medium might be the network of a concrete cable television operator.

However, in most cases, the communication medium in not enough in order to determine what is the intended audience of a communication. In this case, it is possible to use the *recipient* case role in order to specify a *Collective*. Collectives are special concepts that are modelled as defined or enumerated classes. More details about this issue are left to the Implementation chapter 10.

It is important to note that this kind of verbs have a process nature, i.e. they are not temporally associated to a point in time, like most of the previous action verbs. On the contrary, they are associated to a time span where the process takes place. This time span is associated to the verb through two case roles, *start* and *duration*.

Finally, the *origin* case role is used in order to specify the location from which the *Communication* is originated from the legal point of view. This case role is usually required when the communication crosses relevant administrative boundaries, e.g. a satellite communication between different countries. For an overview of the *Communicate* case roles see Table 9.21.

Case role	Range	Cardinality
agent	Person (Natural or Legal)	1..N
theme	Fixation OR Performance	1
result	Communication	1
medium	CommunicationMedium	1
recipient	Collective	0..1
start	e.g. ISO8601	1
duration	e.g. ISO8601	1
location	e.g. ISO3166, URL, etc.	1
origin	e.g. ISO3166, URL, etc.	0..1
...	...	...

Table 9.21: *Communicate* case roles

From these case roles, it is possible to model concrete *Communicate* actions as the one shown in Figure 9.3.4.

Figure 9.16: Model for a *Communicate* action

For practical uses, *Communicate* is too generic. Usually speaking, when communication acts are referred to, the terms used are *Broadcast* and *MakeAvailable*.

Broadcast

When a work is broadcasted, it is made available to a collective through a communication medium. There are two kinds of broadcasting depending on the communication medium. In the first one the medium is the air. A wireless signal is emitted into the air, which can be received by any person, within range of the signal, who possesses the equipment (radio or television receiver) necessary to convert the signal into sounds or sounds and images. In the second one the medium is a cable. A signal is diffused and only persons who possess the required equipment linked to the cables used to diffuse the signal can receive it.

Therefore, for *Broadcast*, the generic *CommunicationMedium* concept can be concretised into *WireMedium* and *WirelessMedium*. This two concepts can be then detailed further into concrete wire mediums, e.g. coaxial cable or optical fiber, or wireless mediums, e.g. GSM or Wi-Fi IEEE 802.11, and even concrete networks. On the other hand, the *result* case role range is concretised to *BroadcastCommunication*.The case roles for *Broadcast* are detailed in Table 9.22.

Case role	**Range**	**Cardinality**
agent	Person (Natural or Legal)	1..N
theme	Fixation OR Performance	1
result	BroadcastCommunication	1
medium	BroadcastMedium (wired or wireless)	1
recipient	Collective	0..1
start	e.g. ISO8601	1
duration	e.g. ISO8601	1
location	e.g. ISO3166, URL, etc.	1
origin	e.g. ISO3166, URL, etc.	0..1
...	...	...

Table 9.22: *Broadcast* case roles

Live Communicate

This is a special kind of *Communicate* where the *theme* is just *Performance*, which is directly broadcasted without any intermediate *Fixation*. The only change respect to *Communicate* is in the *theme* case role, which now refers to *Performance* as it is shown in Table 9.23.

Recording Communicate

This is a special kind of *Communicate* where the *theme* is just *Fixation*, i.e a recording of a previous *Performance*. The only change respect to *Communicate* is in the *theme* case role, which now refers to *Fixation* as it is shown in Table 9.24.

Make Available

Traditionally communication actions related to the communication right were quite passive from the point of view of the communication consumer. The communicator decided when a concrete

Case role	**Range**	**Cardinality**
agent	Person (Natural or Legal)	1..N
theme	Performance	1
result	Communication	1
medium	CommunicationMedium	1
recipient	Collective	0..1
start	e.g. ISO8601	1
duration	e.g. ISO8601	1
location	e.g. ISO3166, URL, etc.	1
origin	e.g. ISO3166, URL, etc.	0..1
...	...	...

Table 9.23: *Live Communicate* case roles

Case role	**Range**	**Cardinality**
agent	Person (Natural or Legal)	1..N
theme	Fixation	1
result	Communication	1
medium	CommunicationMedium	1
recipient	Collective	0..1
start	e.g. ISO8601	1
duration	e.g. ISO8601	1
location	e.g. ISO3166, URL, etc.	1
origin	e.g. ISO3166, URL, etc.	0..1
...	...	...

Table 9.24: *Recording Communicate* case roles

work was broadcasted and the consumers had just to the option to get or not access to the communication medium in order to consume the content. Consumers did not have any alternative to time-shift content consumption without the intervention of recording mediums.

However, communication mediums that allow user interaction have appeared, among which the Internet is the more important one nowadays. The consequence is a particular kind of communication where members of the public access works from a place and at a time individually chosen by them. This kind of communication is called to *MakeAvailable* in the copyright context.

The *result* of such a communication is then an *InteractiveCommunication*. Moreover, one particularity when modelling a *MakeAvailable* is that the *duration* of the process might be greater than the duration of the content that is being made available. Therefore, when a consumer accesses the content, it is not forced to get access to the time point in the content time line that corresponds to the time point from the beginning of the *Communication*. The consumer can then choose its own content time line. Moreover, the medium is a *InteractiveMedium*, which might be wired or not although at this level this distinctions are quite blur, e.g. the Internet.

Finally, there are not *MakeAvailable* actions that have a *Performance* as *theme*, as it is the case with live broadcasts. The *theme* for *MakeAvailble* is just restricted to *Fixation*. This is not possible because a *Performance* cannot be time-shifted and converted into an *InteractiveCommunication* without an intermediate *Fixation*. The case roles for *MakeAvailable* are detailed in Table 9.25.

Case role	**Range**	**Cardinality**
agent	Person (Natural or Legal)	1..N
theme	Fixation	1
result	InteractiveCommunication	1
medium	InteractiveMedium	0..1
recipient	Collective	0..1
start	e.g. ISO8601	1
duration	e.g. ISO8601	1
location	e.g. ISO3166, URL, etc.	1
origin	e.g. ISO3166, URL, etc.	0..1
...	...	...

Table 9.25: *MakeAvailable* case roles

These are the only differences respect to *Broadcast*. The important difference between *MakeAvailable* and *Broadcast* arises at consumption time, when the consumer accesses a communication. More details about this aspect are given in the Usage Actions section 9.3.5.

9.3.5 Usage Actions

These are the verbs related to the consumption of copyrighted content.

Access

This is the action that consumes the communication of a *MakeAvailable*, i.e. an *InteractiveCommunication*. Consequently, it is an interactive action from the point of view of the consumer, who is able to determine the time line of the content. Usually, the content is consumed from its beginning and from the time point when the content is accessed. Moreover, the user is usually free to pause, resume, forward and rewind the content. However, all these issues are specific to the technological means used to make the content available, e.g. internet video on-demand streamers. It is important to note that this same features can be enjoyed with non-interactive communications but a recording system at the consumers place is then required.

The *Access* verb is defined by the case roles shown in Table 9.26. It is important to highlight that the *theme* is restricted to *InteractiveCommunication*, i.e. the result of a *MakeAvailable* communication action. On the other hand, a part from the *pointInTime* when the *Access* takes place, it is possible to define an intended *duration* for the *Access*. Finally, it is also common to enjoy interactivity at the level of the characteristics of the accessed content. For instance, the consumer can choose the resolution of a video, the audio signal bit rate encoding of a song, etc. This is specified by the *manner* case role that defines some *ContentCharacteristics* of the accessed content, which are detailed in the Domain Specific Ontologies section 9.4.2.

Figure 9.17 shows an example of a concrete *Access* action.

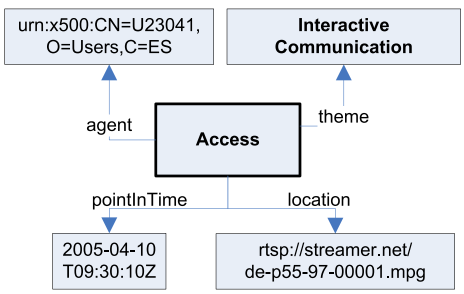

Figure 9.17: Model for an *Access* action

Case role	Range	Cardinality
agent	Person (Natural or Legal)	1..N
theme	InteractiveCommunication	1
pointInTime	e.g. ISO8601	1
start	e.g. ISO8601	0..1
duration	e.g. ISO8601	0..1
medium	InteractiveMedium	0..1
location	e.g. ISO3166, URL, etc.	1
manner	ContentCharacteristic e.g bit rate, resolution, etc.	0..N
...	...	...

Table 9.26: *Access* case roles

Tune

This is the action that consumes the communication of a *Broadcast*. Consequently, it is a non-interactive action from the point of view of the consumer. The user just tunes the *medium* where the content is broadcasted from an appropriate *location*. The *theme* is thus a *BroadcastCommunication*. For the rest of the case roles it is like an *Access*, as it is shown in Table 9.27.

Case role	Range	Cardinality
agent	Person (Natural or Legal)	1..N
theme	BroadcastCommunication	1
start	e.g. ISO8601	1
duration	e.g. ISO8601	0..1
medium	BroadcastMedium (wired or wireless)	1
location	e.g. ISO3166, URL, etc.	1
manner	ContentCharacteristic e.g bit rate, resolution, etc.	0..N
...	...	...

Table 9.27: *Tune* case roles

Attend

This is the action corresponding to the consumption of a *Perform* in a *PublicPlace*. The *theme* is a *Performance* which result from a *Perform* that is held in the same *PublicPlace*. Therefore, there is a time sharing between the *Attend* and the corresponding *Perform*. The complete case roles are detailed in Table 9.28.

Case role	Range	Cardinality
agent	Person (Natural or Legal)	1..N
theme	Performance	1
pointInTime	e.g. ISO8601	1
location	PublicPlace	1
...	...	...

Table 9.28: *Attend* case roles

Buy

This action is the counterpart of the *Sell* action. It is not required as it just reverses the corresponding *Sell*. It is included here in order to clarify this point.

9.3.6 Commercial Actions

These are the verbs related to actions related to the commercialisation and negotiation of commercial terms of copyrighted content.

Transfer

The *Transfer* action is a very generic one that corresponds to one of the verb patterns in the left side of the Case Roles Table 9.7. It is used to model the economic interchanges inherent to a commercial system, specially monetary transfers required as compensations for rights and usage licenses of copyrighted content. Additionally, many copyright legal framework grant the possibility to transfer any of the economic rights in addition to the possibility to license specific actions. This means that the original holder of the right transfers it to another person or entity, which becomes then the right holder. Therefore, rights can be also the *theme* of a *Transfer*.

The basic *Transfer* is based on the case roles shown in

Table 9.28.

Case role	Range	Cardinality
agent	Person (Natural or Legal)	1..N
theme	Thing	1
recipient	Person (Natural or Legal)	1..N
...	...	...

Table 9.29: *Transfer* case roles

An example of *Transfer* is shown in Figure 9.18 in the next section. This basic *Transfer* can be then extended with other case roles in order to model many different compensation methods, e.g. fee flat, prepay, royalties, fee per use, fee per interval, fee metered or fee per use prepay. Moreover, more than one *Transfer* can be used in order to model complex compensation methods.

Offer

In order to cope with content negotiation processes, the Copyright Ontology includes some negotiation specific concepts, although in a very limited way as this is not the purpose of the ontology. The negotiation process that is modelled is based on an initial *Offer*, followed by zero or more *Counteroffers* and, at the end, an eventual *Agreement* if the negotiation is not abandoned before.

The offer defines the concrete *agent* that performs the action and in some cases the *experiencer* as a generic class that includes the intended audience of the offer, thus the potential person that may *Counteroffer* and *Agree* on the terms of *Offer*. The more general range for *experiencer* is LegalPerson, i.e. when the intended audience is any legal person. The *theme* specifies what is offered, generically anything. The case roles for *Offer* are detailed in Table 9.30.

Figure 9.18 shows an example of a concrete *Offer* action together with the associated offered *theme* and the required *condition*, a *Transfer* .

Counteroffer

A *Counteroffer* follows a previous *Offer* or *Counteroffer*. They are similar to *Offer*, the *agent* points to the counterofferer, some members of the original *Offer* intended audience or of the original offerers. In addition, the previous *Offer* or *Counteroffer* in the negotiation chain, i.e. the one for which the

Case role	**Range**	**Cardinality**
agent	Person (Natural or Legal)	1..N
theme	Thing	1
recipient	Person (Natural or Legal)	1..N
pointInTime	e.g. ISO8601	1
start	e.g. ISO8601	0..1
duration	e.g. ISO8601	0..1
...	...	...

Table 9.30: *Offer* case roles

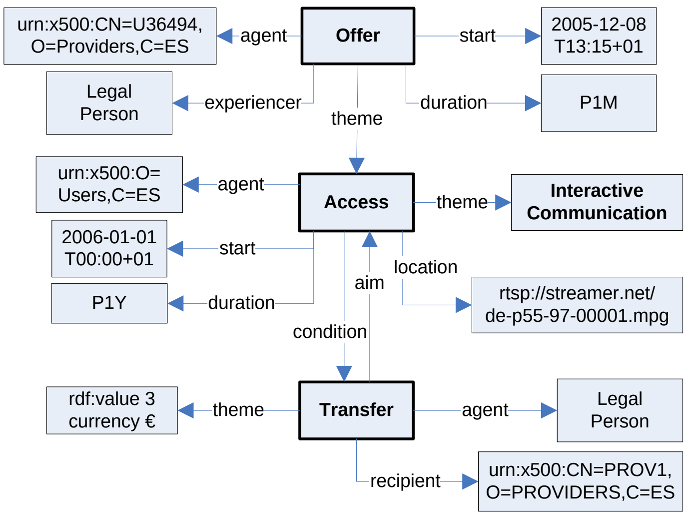

Figure 9.18: Model for an *Offer* action

current one is a *Counteroffer*,is pointed by the *origin* case role. The case roles for *Counteroffer* are detailed in Table 9.3.6.

Case role	**Range**	**Cardinality**
agent	Person (Natural or Legal)	1..N
theme	Thing	1
recipient	Person (Natural or Legal)	1..N
pointInTime	e.g. ISO8601	1
start	e.g. ISO8601	0..1
duration	e.g. ISO8601	0..1
...	...	...

Table 9.31: *Counteroffer* case roles

Agree

An *Agree* signals the end of a successful negotiation process and it enables that its theme may be fulfilled, i.e. it is possible to do what is agreed. An *Agree* is identical to the last *Counteroffer* in the negotiation chain, or identical to the initial *Offer* if it has not been negotiated but directly accepted. The only difference is that it does not contain an *experiencer* case role. Instead, all the parties that hold the agreement are *Agreeagents*. The case roles for *Agree* are detailed in Table 9.3.6.

Case role	**Range**	**Cardinality**
agent	Person (Natural or Legal)	1..N
theme	Thing	1
pointInTime	e.g. ISO8601	1
...	...	...

Table 9.32: *Agree* case roles

9.3.7 Moral Rights Actions

These are the verbs related to actions governed by moral rights.

Attribute

The *Attribute* action corresponds to the right of the author to claim authorship of one of its *Works*. The author is the *agent* and *experiencer* of the action andthe *theme* is the *Work*. Moreover, there are some circumstances that may require that a third party attributes the *Work* to the *Author*. In this case the *agent* is a third party, i.e. a *Person* different from the author. Examples of circumstances that require third party attribution are when a *Quote* is performed or when it is the condition of an *Agree*, e.g. Creative Commons Attribution License[1].

Oppose to change

Due to the integrity moral right, the author of a work can exercise a *OpposeToChange* action. This action allows the author to oppose to a concrete change to the *Work*, which is specified by a *Derive* action, or more concretely an *Adapt* or a *Translate*. The author is the *agent* and the *theme* is the *Derive* action that the author opposes to.

Disclose

This act corresponds to the *Manifest* action. In some legal systems the author has the moral right to block that a *Work* gets public and therefore it cannot be manifested from the legal point of view. The author is the *agent* and the *theme* is the *Work* that is disclosed.

Withdraw

Once a *Work* has been manifested, in some legal systems the author has the right to retire it from the public scene. This action is performed by the *Withdraw* action. The author is the *agent* and the *patient* is the *Work* that is retired.

9.3.8 Exceptional Actions

Certain end-user acts normally restricted by copyright may, in circumstances specified in the law, can be done without the authorization of the copyright owner. These exceptions to copyright should be considered as end-user privileges and not rights. The following actions are special cases for some of the previously defined ones. Under their special conditions, they can be performed

[1] http://creativecommons.org/licenses/by/2.5

without authorisation, i.e. an *Agree* that enables them. However, this does not mean that they are free, in some cases a economic compensation might be required or is implicit, e.g. there might be levies on digital recording equipment and media.

Quote

A *Quote* is a limited extent *Copy* action of a source protected *Work*, which is clearly mentioned so it has a *condition* case role that points to a *Attribute* action for the quotation to the author of the *Work*. The *theme* of the *Quote* is a piece of limited size of the quoted *Work*. Pieces are related to the *Work* through a *isPartOf* relation and determining that they are of limited size lies outside the scope of the Copyright Ontology.

Educational Act

An *EducationalAct* is any *Copy*, *Communicate* or *Perform* action that has a *Educational* or *Research* purpose as the range of its *aim* case role.

Inform

An *Inform* action is like an *EducationalAct* but a purpose that is *Informative* is its *aim*.

Official Act

An *OfficialAct* is any copyright governed action that categorised this way.

Private Copy

This is like any other *Copy* action with the particularity that it has an *aim* case role that points to a *Private* purpose. The *Instance* resulting from the action cannot be the *theme* of any other economic rights action or usage action with an *agent* different from the one that realises the *Copy* or any other agent outside its family or friends groups.

Parody

A *Parody* action is like an *EducationalAct* or *Inform* but it has a *Parody* or *Caricature* purpose as the range of its *aim* case role.

Reproduce Temporally

It is like any other *Copy* action with the particularity that its *aim* case role has a *Temporal* purpose as range.

9.3.9 Other Actions

Abstract

This is the action by which the *Work* underlying a *Manifestation*, *Performance*, *Fixation*, *Communication* or *Instance* is recognised. Any agent performs this action, the *theme* is a *Manifestation*, *Performance*, *Fixation*, *Communication* or *Instance* and the *result* is a *Work*.

9.4 External Concepts

Some concepts that are external to the copyright conceptual model have been employed so far. Some of them come from external upper ontologies and were used in order to establish the conceptual base in the Creation Model section 9.1. Other come from domain specific ontologies or have been just defined in the context of the Copyright Ontology.

9.4.1 Upper Ontologies

To conclude, the contributed copyright ontology is enriched with general concepts for time, space, tools, parthood, etc. They are taken from upper level ontologies, which define general concepts. For the moment, we have considered some upper ontologies, detailed in the Upper Ontologies section 2.9.1: IEEE SUMO [84], DOLCE [41] and LRI-Core [16]. Our intention is to make general concepts from upper ontologies interchangeable and make alignment of the copyright ontology to all these top ontologies possible.

The following subsection point out some connections from the Copyright Ontology to these upper ontologies. This relations are quite rough as there might be many incompatibilities among the upper ontology concepts a given Copyright Ontology concept is mapped to.

Abstract

It has been used in the Copyright Ontology as the place to hang the *Work* concept and defined as something that cannot exist at a particular place and time without some physical encoding or embodiment. It is similar to "Mental Concept" or "Non-agentive Social Object" in DOLCE. In the context of LRI-Core, it can be related to "Mental Concept" and, in the context of IEEE SUMO, to "Abstract".

Object

It corresponds roughly to the class of ordinary objects and it is related in the Copyright Ontology to *Manifestation*, *Communication* and *Instance*. This concept can be related to the "Physical-Endurant" concept in DOLCE, i.e. a continuant that has spatial qualities. It can be also associated to "Physical-Object" in LRI-Core and "Object" in IEEE SUMO.

Process

It is something that happens and has temporal parts or stages. *Process* is used in the Copyright Ontology as the place holder to hang *Performance* and *Communication*. It can be related to the "Physical-Perdurant" concept in DOLCE, i.e. an occurrent that has spatial qualities. Moreover, it can be connected to "Physical-Process" in LRI-Core and "Process" in IEEE SUMO.

Person

In the context of the Copyright Ontology, this concept stands for legal person. It can be related to more general concepts like "Social Agent" in DOLCE or "Agent" in IEEE SUMO.

isPartOf

This is a fundamental relation from the ontological point of view. It is used to specify compound works, i.e. compilations.

It can be connected to the "part_of" relation in LRI-Core or to the "part-of" a-temporal parthood relation in DOLCE.

9.4.2 Domain Specific Ontologies

Some specific concepts that lay outside the scope of the copyright ontology have been reused from external domain ontologies. A short list of them follows. They are very specific so the Implementation chapter 10 contains more details about how they have been connected to the implementation of the Copyright Ontology.

- **Collective**
- **Time**
- **Location**
- **Content Characteristic**
- **Currency Measure**

On the other hand, there are other concepts that are used in the previous copyright conceptual model but are very hard to reuse from other ontologies. This difficulty is due in some cases to the excessive detail of the corresponding ontologies or to the lack of ontologies for these concepts. They are just named in the context of the Copyright Ontology but no more detailed definitions are provided.

- **Communication Medium**: it has two defined subconcepts, i.e. *BroadcastMedium* and *InteractiveMedium*.
- **Purposes**: some specific purposes have been used, i.e. *Temporal*, *Informative*, *Parody*, *Caricature*, *Private*, *Educational* or *Research*.
- **Public Place**

Chapter 10

Implementation

The conceptual model detailed in the Conceptualisation chapter 9, is just a conceptual model. In order to get a formal and explicit formalisation of this conceptualisation in order to get the corresponding ontology. This step can be seen as a combination of the formalisation and implementation phases as detailed in the Methontology section 7.2.

This two phases have been combined in this chapter, and it is named just "Implementation", because, although the formalisation tasks are also considered, this chapter gets into much more detail in the implementation part. The formalisation is much more mechanic while the implementation is more tricky due to the special implementation approach that has been considered in this work.

It is important to note that the implementation approach documented in this section is just one of the possible implementations of the copyright conceptual model. Moreover, it does not take into account formal notions of right, modal operators, etc. On the other hand, the objective has been to produce an ontology that can be easily put into practice with tools that easily scale to great amounts of data and that can be integrated in production systems.

Therefore, in order to avoid tractability and decidability problems, the extensive use of DL (Description Logic) reasoners has been one of the main driving forces of this implementation. The characteristics of DL reasoners that justify this decision are detailed in the Description Logics section 2.7. A Semantic Web implementation of DL is employed: the Decription Logic version of the Web Ontology Language (OWL-DL) [23].

This decision has produced a very naive formalisation of right and deontic operators (obligation, permission and prohibition) that are just valid in the context of this implementation of the copyright conceptual model, i.e. the OWL-based Copyright Ontology. This naive approach has been also extended to other aspect of the ontology with one idea in mind, to directly take profit from the classification functionalities of DL reasoners in order to implement license checking.

This includes to check if a particular use is allowed by some agreement, if there is any offer that might allow it, if the required compensation has been fulfilled, if the agent of the usage action is a member of the intended users class, etc. More details are given in the Description Logic Mechanisms section 10.2.

Another objective of this implementation is to reuse many things from other sources. A detailed account of the sources that have been employed in order to complement the Copyright Ontology with practical means to deal with time, locations, etc. is presented in the Reused Terms section 10.4.

To conclude, it is important to note that the main part of the implementation work is not visible in this chapter but in the resulting OWL Copyright Ontology, which is available from the Copyright Ontology web site[1]. Moreover, this chapter ends with a Example Scenarios section 10.5 that documents practical uses of the ontology and for more details the Semantic DRM System section 11.4 describes how the ontology works in the context of a Semantic Digital Rights Management System.

10.1 Introduction

The previously detailed models, i.e. Creation, Rights and Action Model, plus the required concepts from external ontologies or just defined to build up our ontological framework for copyright. This conceptual model has been implemented using Semantic Web ontologies and rules languages.

The main objective has been to provide a straightforward and efficient implementation. In order to do that, in the context of web ontologies, we have chosen OWL-DL [87]. OWL-DL is a Web Ontology language that is also a Description Logic (DL). Therefore, it can be directly fed into DL classifiers, which are specialised logic reasoners that deal with class definitions and instances. They guarantee tractability and decidability for class subsumption checking and instance classification.

DL classifiers are used in order to automatically check copyright-governed events against copyright rights and the action patterns specified in copyright situations, agreements and offers. This facilitates checking if a particular action, once modelled as an event, is allowed or not. It is even possible, if the action is not disallowed, to look for offers that grant action patterns that would

[1] http://rhizomik.net/ontologies/copyrightonto

enable it, once an agreement is reached, etc.

DL classifiers can be directly reused so there is no need to develop ad-hoc applications to perform this function. The more complex behaviours that cannot by captured using OWL-DL are modelled using Semantic Web rules, described in the Semantic Web Rules section 4.8.6. However, this is just the implementation at the ground level. All this must be complemented with a metalevel that implements about the deontic aspects, cf. the Logic Types section 2.6.1, that are implicit in the conceptual model.

This metalevel guides the DL checks and rules executions that have to be performed in order to capture the semantics of the implicit obligations, permissions and prohibitions, as it is detailed in the Description Logic Mechanisms section 10.2. For instance, the metalevel performs the DL classification of the usage instance model under test and then check that there is at least on usage pattern that subsumes it. Additionally, the usage pattern must be the *theme* of an Agree and the *condition* of the usage pattern must be fulfilled, i.e. it is checked that there is an instance subsumed by it in order to conclude that it has been already satisfied.

The metalevel can be implemented also using Semantic Web rules. However, as a first approximation, it is going to be implemented programmatically as part of a Semantic DRM System.

10.2 Description Logic Mechanisms

An extensive use of Description Logic mechanism is made in order to facilitate the implementation of DRM systems that take profit from the Copyright Ontology. The first consequence of this approach is that it seems as all is modelled as instances inside OWL classes inside OWL classes, etc. It is quite clear that the concepts in the Creation Model section 9.1 must be modelled as classes with the associated OWL restrictions for the corresponding relations. The tricky part is how to employ classes with the other models.

First of all, the rights in the Rights Model section 9.2 are modelled as classes of actions that contain the action shown as the ones governed by the right as subclasses, together with other rights that might appear below them in the rights hierarchy. For instance, the *CommunicationRight* has *Communicate*, *Broadcast*, *Retransmit* and *MakeAvailable* as subclasses. However, *Communicate* is the only direct subclass action of the *CommunicationRight*, the other ones are indirect subclasses. They are subclasses of the rights that are subclasses of *CommunicationRight*.

Consequently, all the actions in the Action Model section 9.3 are also modelled as classes, which are subclasses of the corresponding class that models the right that governs that class of actions. With this weird connection between rights and actions it is possible to reduce checking if an action is authorised by a given right to just check if the right class subsumes the action class.

In addition, agreements and offers are modelled as classes and they have a theme relation that points to the granted action pattern. This pattern is also modelled as a class and associated to the *Agree* or *Offer* class using an "owl:allValuesFrom" restriction. Patterns correspond to the event constructions used in the Action Model, i.e. they are vased on a verb and some case roles with specific ranges. Therefore, events are also modelled as classes.

It is important to note that not just the generic events presented in the Action Model are modelled as classes, even the specific ones used as patterns of the actions granted by an agree or offered by an offer. For the more specific actions the "owl:hasValue" restriction is used so it is even possible to build patterns that define some concrete instance as the range of one of its case roles, e.g. a concrete user or a specific location. Therefore, very concrete events can also be modelled as classes and used afterwards as patterns or models for the actual action an user is trying to perform.

This implementation approach is exemplified with the conceptual model of an offer shown in Figure 9.18 in the Conceptualisation chapter. 9 The more important changes in comparison to the original conceptual model are due to the OWL-DL limitations required for its tractability. Classes are not directly related to other classes and instances. In order to model OWL classes relations, the "owl:Restriction" primitive is used, one for each relation that is specified with "owl:onProperty".

The restricted values for the property are specified with "owl:allValuesFrom" or "owl:someValuesFrom" when the range is a class and "owl:hasValue" when the range is limited to an instance. For instance, if the class is related through the *duration* property to the instance "P1M", it is implemented as an "owl:Restriction" "owl:onProperty" *duration* "owl:hasValue" "P1M".

In the Figure, the same arrows that appear in the conceptual model are used now to represent each "owl:Restriciton". The arrow tail points to the restricted class, i.e. the class that is a "rdfs:subClassOf" the restriction. The arrow head points to the values to which the property is restricted to take. The arrow also specifies the name of the property and the kind of restriction, i.e. "allValuesFrom", "someValuesFrom" or "hasValue".

At this point, the user actions has to be checked against a repository of agreements in order to see if the action is allowed. Therefore, there is an implicit deontic operator permission that is implemented using the DL reasoner. The class modelling the action is classified against the set of agree classes and their associated usage patterns, i.e. their themes. Then, the user action is considered to be potentially permitted if its classified as a subclass of any of the usage patterns included in the agreements repository.

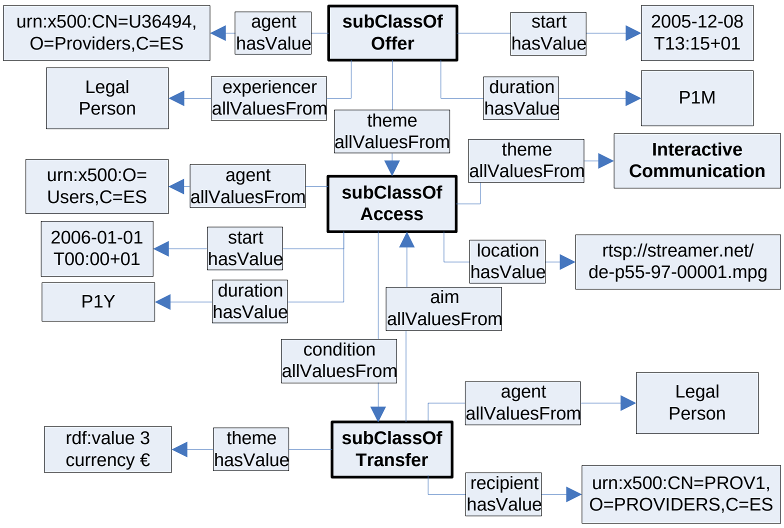

Figure 10.1: OWL-DL implementation of the *Offer* conceptualisation in Figure 9.18

It is not effectively permitted yet because two more conditions have to be checked. First, in order to be a permission, the usage patterns where the user action has been classified must be the theme of an *Agree*. On the other hand, if it is the theme of an *Offer*, the idea is to trigger a negotiation process that might eventually make the action permitted if an agreement is reached.

In addition to the previous check, the usage patterns usually specify an obligation for the user in order to perform the action. This is specified by the condition case role. It can be viewed as an implicit obligation deontic operator. Therefore, in this case, the second condition is to check that the obligation has been fulfilled. This is also done using DL mechanisms. The obligation range is also a class that models a Transfer event. The obligation is satisfied if the is a resource in the repository that has been classified as an instance of the obligation pattern.

To conclude, it is also important to note that it is possible to model prohibitions. This is done using "owl:complementOf" in order to get negated classes, e.g. a class defined as "complement of location equal to DVD Zone 3", and then intersected with the usage pattern with "owl:intersectionOf". More details about how the DL mechanisms are put into practice are available in the Example Scenarios section 10.5.

10.3 Semantic Web Rules

In order to implement the rules and axioms included in the conceptual model, Semantic Web rules have been used. This enables an easy integration of the Description Logic mechanisms and the rules. First, the rules have been translated to KIF, as it is shown in the Table 10.1. Then, the Sweet-Rules[2] KIF to SWRL translator has been employed to generate the corresponding Semantic Web rule. It has been necessary to employ the more expressive version SWRL-FOL [89] as the conceptual model rules are First Order Logic.

```
(forall (?v Manifest)
(exists ((?mr MoralRights)(?er EconomicRights))
(forall ((?p Person)(?m2 Manifestation)(?w Work)(?m Manifestation)(?t Time)
(=> (and (agent ?v ?p) (theme ?v ?w) (result ?v ?m) (pointInTime ?v ?t)
(not (isManifestationOf ?m2 ?w))))
(and (agent ?mr ?p) (agent ?er ?p) (essence ?mr ?w) (essence ?er ?m)
(start ?mr ?t) (start ?er ?t) (isManifestationOf ?m ?w)))
)))))
```

Table 10.1: KIF rule to assign author rights

10.4 Reused Terms

Implementation decisions for some special concepts that lie out of the copyright ontology domain but are required in order to build rights expressions.

10.4.1 Location

Many methods depending on the level of detail required. However, in the copyright context, the more common method to specify location is based on country codes as they define the natural boundaries of legal systems. The ISO 3166[3] country codes are used as the standard mean to specify countries. Fine-grained location names are available from the Getty Thesaurus of Geographic Names[4]. For even more concrete locations it is possible to use latitude-longitude coordinates as proposed by the W3C SWIG[5].

This is the method used in order to determine physical locations. However, when working

[2] http://sweetrules.projects.semwebcentral.org
[3] http://www.iso.org/iso/en/prods-services/iso3166ma/02iso-3166-code-lists/list-en1.html
[4] http://www.getty.edu/research/tools/vocabulary/tgn/
[5] http://www.w3.org/2003/01/geo

in the virtual space provided by the Internet and the World Wide Web, specific location definition means are required for this non-physical spaces. In this context the more appropriate location method is URL (Uniform Resource Locator) specified by the RFC 1738[6] recommendation.

10.4.2 Time

ISO 8601[7] is used for time points, time intervals, durations and recurring intervals. Time points include date, time and combined date and time expressions based on the "YYYY-MM-DDThh:mm:ss±hh:mm" expressions, e.g. "2005-11-25T12:00:00+01:00". If no so much precision is required the more fine grained elements of the expression can be omitted, e.g. seconds or seconds and minutes, etc.

Durations are represented by the format "PnYnMnDTnHnMnS". For instance, "P3Y6M4DT12H30M0S" defines a period of three years, six months, four days, twelve hours, thirty minutes, and zero seconds. Elements may be omitted if their value is zero.

Time intervals specify an amount of time. They may be specified in four ways:

1. Start and end, such as 2002-03-01T13:00:00Z/2003-05-11T15:30:00Z
2. Start and duration, such as 2002-03-01T13:00:00Z/P1Y2M10DT2H30M
3. Duration and end, such as P1Y2M10DT2H30M/2003-05-11T15:30:00Z
4. Duration only, such as P1Y2M10DT2H30M

Finally, Repeating intervals are formed by adding "Rn/" to the beginning of an interval expression, where "R" is used as the letter itself and "n" is replaced by the number of repetitions. Leaving out the value for "n" means an unbounded number of repetitions. So, to repeat the interval of "P1Y2M10DT2H30M" five times starting at "2002-03-01T13:00:00Z", use "R5/2002-03-01T13:00:00Z/ P1Y2M10DT2H30M".

10.4.3 Collective

As it has been pointed out in the Description Logic Mechanisms section 10.2, collectives are also modelled as OWL classes. They can be a class defined using OWL restrictions that can be necessary or both necessary and sufficient. Moreover, they can be enumerated classes. With this approach, it is possible to define the intended audience of an usage pattern as a collective class,

[6] http://www.ietf.org/rfc/rfc1738.txt
[7] http://www.iso.org/iso/en/prods-services/popstds/datesandtime.html

i.e. "owl:allValuesFrom" the collective class. Afterwards, it is possible to implement checking if the usage is granted to a concrete user as a simple DL classification.

For instance, the usage pattern defines the intended audience of the grant, i.e. the range of the agent case role of the action. The intended audience is all members of a given university. Therefore, the class for the collective is defined as a class with a necessary and sufficient restriction on the member property "owl:hasValue" the given university. Afterwards, any user action that has to be checked against this pattern would be enables as long as the user is a member of the university, an consequently is classified in the collective class.

10.4.4 Content Characteristic

When interactively accessing content, it might be possible to choose some of the characteristics of the content. This generic class is a place holder for specific content characteristics that lie outside of the scope of the Copyright Ontology. Therefore, the recommendation is to reuse content characteristics from multimedia description ontologies. The most complete multimedia description framework is MPEG-7. It is defined by some XML Schemas but it is also available as an OWL ontology [42].

10.5 Example Scenarios

10.5.1 Streaming Server Scenario

The use of DL classifiers for digital rights management in the context of the copyright ontological framework can be exemplified with the following scenario:

The initial situation is: "USER1 is trying to access a given video stream from a given streaming server at 9:30:10 UTC on 2005-04-10". The streaming server implements digital rights management so it inquires the license manager if the current usage instance is permitted. In order to do that, the streamer models this usage as shown in Figure 10.2, and sends it to the license manager, e.g. as a RDF/XML serialisation.

The license manager contains licenses modelled using the same approach, among others the one shown in Figure 10.3. This license grants a usage pattern for a creation located at the streaming server that can be performed by a class of agents for a given period of time starting on a given date. Moreover, the license manager has additional metadata stating that USER1 is an instance of the pattern users class.

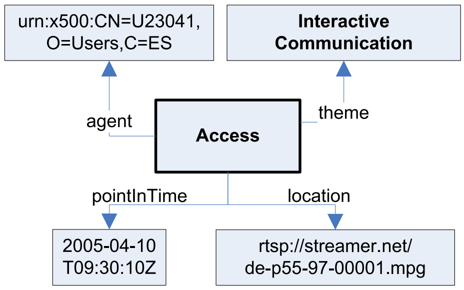

Figure 10.2: Usage instance modelled by the streaming server

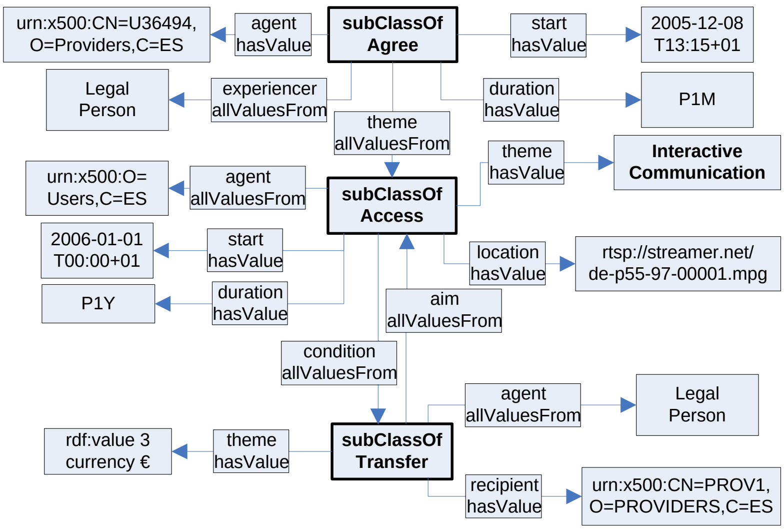

Figure 10.3: Use license model defining permitted usage pattern and condition

The license manager checks if there is any license that grants a usage pattern that subsumes the usage instance. This can be performed easily and efficiently using a DL classifier. However, before that, it is important to note that usage patterns define the time interval using a start time and duration, while the usage instance defines a time point. In order to check if the time point is included in the time interval, we must use a DL classifier capable of dealing with custom datatypes reasoning [88]. Then, the time interval is translated to a real interval, i.e. *pointInTime* is greater or equal than 20050401 and smaller or equal than 20060401, and the time point to a real, i.e. *pointInTime* is equal to 20050410.093010.

After applying the previous adjustment, subsumption is computed. The usage might be classified in one or more usage patterns. In this case it is tested if the usage pattern is the theme of an Agree concept. Then, if there is an instance of the condition, i.e. it is satisfied, the license manager tells the streaming server that the use is authorised. Otherwise, the use is not authorised.

This is a simple scenario for illustrative purposes. It could be extended in many ways. For instance, if the usage pattern is the theme of an offer, another possibility is to recommend the user the possibility to negotiate it in order to arrive to a new agreement. From this point, it can be connected to negotiation architectures [26, 48] which have been included in the contributed Semantic DRM System 11.4.

10.5.2 Withdrawal Right Scenario

In order to show the capabilities of the copyright modelling framework detailed in the previous section, we are going to show how it can be used to model a complete copyright scenario. We have chosen a quite uncommon one because it does not deal with exploitation-oriented aspects.

The scenario is about moral rights, concretely the withdrawal right. In this scenario, the author exercises one of its inalienable rights to retire one of its works from the public scene, as he does no longer consider that it represents his personality. The whole scenario is considered; from the moment when the author creates the work, and correspondingly acquires full rights on his creation, until the consequences of its withdrawal. The scenario steps are detailed next.

Creation and Acquisition of Moral Rights

The steps leading to the acquisition of the moral rights that grant the author the possibility to exercise a withdraw of its work are captured by the manifest action that takes the first manifestation of the work into existence and the corresponding rule that assigns to the author the moral rights. The event and the rule are like those presented in the Manifest section 9.3.3 and the resulting situation is shown in Figure 10.4.

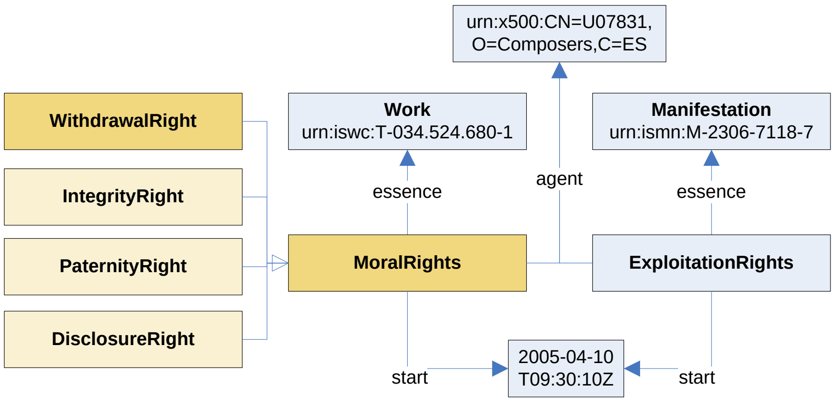

Figure 10.4: Detailed moral rights situation

Withdraw and Compensation

From the previous step, the author is now authorised to withdraw his work as it is shown in Figure 10.5. The withdraw action is subsumed by the Withdrawal Right. However, this act will have its consequences. Generally speaking, the consequence is that he should compensate the third parties with which he has established exploitation agreements for the economical damages this act may impose them.

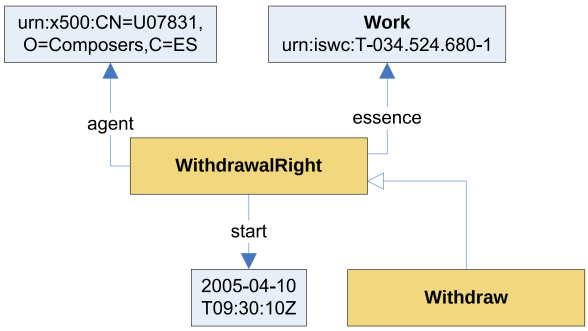

Figure 10.5: Withdraw enabled by withdrawal right

These compensations can be explicitly anticipated in the previous agreements or inferred from additional rules or external systems when the withdraw is performed. Figure 10.6 shows one exploitation agreement for communicating the work in exchange of a compensation to the author, which is named *Transfer A* in the figure. There is an additional provision in this agreement, another transfer from the author to the other party that is conditioned to the exercise of the withdraw act of the work, *Transfer B*.

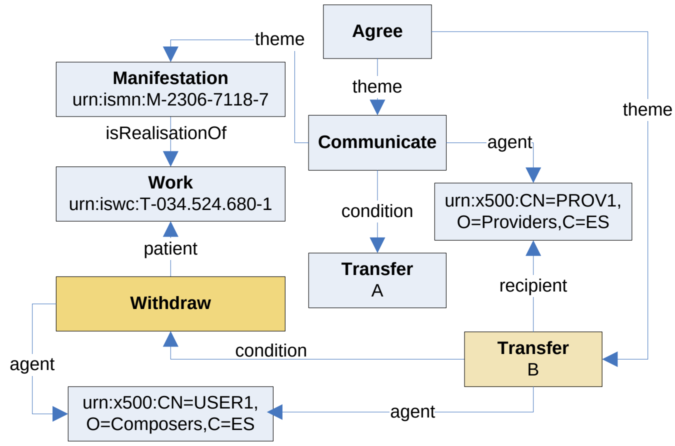

Figure 10.6: An agreement with withdraw compensation provision

Chapter 11

Evaluation

This evaluation chapter has been conceived as the place to show all that has been done in order to put the contribution of this work into practice and validate it. The more direct way of evaluating the contribution is using ontology evaluation tools like the ones provided by some ontology editors or reasoners. This kind of evaluation has been performed during the ontology formalisation phase, which has not been documented in this work as it has been automatically performed by the ontology modelling tools that have been employed.

There are also indirect ways of evaluating it. The approach that has been taken for this is to take profit from the XML Semantics Reuse methodology in order to generate ontologies for the main rights expression languages that are based on XML Schemas: MPEG-21 REL and ODRL.

The same has been done for MPEG-21 RDD, which is not an XML Schema but a ontology. However, it is not defined using a formal ontology language. Therefore, a method to map it to a web ontology has been also developed and it has produced also a formal ontology for the RDD.

Once all this initiatives are available as web ontologies, it has been possible to put the Semantic Web approach into practice with them. The benefits of moving them to the Semantic Web are shown. Therefore, there is an additional contribution independent from the main contribution, i.e. the Copyright Ontology. It is to apply the Semantic Web to other REL initiatives.

Moreover, another consequence of moving all this initiatives to web ontologies is to facilitate validating the Copyright Ontology. Once they are also in OWL form, it is easy to try to map them to the Copyright Ontology. It can be then checked if all the concepts in this ontologies have an anchor point in the Copyright Ontology where they can be mapped.

Although all the previous initiatives have been mapped to web ontologies and used to implement semantics-enabled applications that show the Semantic Web approach benefits, their mappings to the Copyright Ontology have not been completed.

Then, it remains future work to do an extensive evaluation of the Copyright Ontology in relation to other REL initiatives. However, to the extent that this mapping has been already implemented, it can be concluded that the Copyright Ontology provides a base framework where this initiatives can be plugged-in using Semantic Web tools based on ontology primitives for concept inclusion and equivalence and Semantic Web rules.

Finally, the Copyright Ontology has been put into practice in the context of a semantics-enabled DRM system called NewMARS. This system has been developed using a knowledge-oriented approach. The more important part of the system is the knowledge layer, which is composed by the Copyright Ontology and other multimedia description ontologies.

On top of this ontologies layer, the whole system has been implemented based on semantic metadata. Semantic metadata is present from storage access, which is performed by semantic queries, through retrieval and metadata integration, which is done based on RDF graphs, to metadata rendering and user interaction, which is based on HTML rendering of RDF metadata and semantics-enabled HTML forms.

11.1 ODRL Ontology

In order to move Digital Rights Management to the Internet, a common rights expression language is needed. ODRL (Open Digital Rights Language) is one of the proposed solutions. It is based on a XML language and thus it just formalises the language syntax, while language semantics are specified informally.

Actually, ODRL seems quite complete and generic enough to cope with such a complex domain. However, the problem is that it has such a rich structure that it is difficult to implement. In our opinion, it lacks formal semantics that would help ODRL applications development.

As the application context is the Web, the approach to formalise ODRL semantics is based on semantic web ontologies. Firstly, ORDL has been moved to the Semantic Web space using XML Schema to OWL and XML to RDF tools detailed in the XML Semantics Reuse section 7.3. This provides some simple semantics.

In order to refine them, the resulting ODRL ontologies have been connected to the copyright ontology described in the contribution part of this work. The interrelation of the copyright ontology with the ontologies resulting from the XML Schema to OWL mapping enables semantics-aware

ODRL applications that benefit from semantic queries.

This contrasts with the difficulties that emerge from the use of syntactic queries when the information space is as complicated as in the DRM field. Moreover, specialised reasoners can be used for license checking and retrieval. All these advantages have been propagated to ODRL thanks to this mapping.

11.1.1 Introduction

The amount of digital content delivery in the Internet has made Web-scale Digital Rights Management (DRM) a key issue. Traditionally, DRM Systems (DRMS) have deal with this problem for bounded domains. However, when scaled to the Web, DRMSs are very difficult to develop and maintain. The solution is interoperability of DRMS, i.e. a common framework for understanding that defines a shared rights expression languages and its associated vocabulary.

ODRL (Open Digital Rights Language[1]) [64] is one possible approach to that. As it has been detailed in the State of the Art ODRL section 5.4, it is a XML language defined by two XML Schemas. The first XML Schema, called EX-11, defines the language syntax and a basic vocabulary. The second XML schema is called DD-11 and it contains the data dictionary. It provides the complete vocabulary with textual definitions and a lightweight formalisation of the vocabulary terms semantics as an XML Schema.

ODRL seems quite complete and generic enough to cope with such a complex domain. However, the problem is that it has such a rich structure that it is difficult to implement. It is rich in the context of XML languages and the "traditional" XML tools like DOM or XPath. There are too many *attributes*, *elements* and *complexTypes* to deal with, as it is shown in Table 11.1.

	ODRL	
Schema	EX-11	DD-11
xsd:attribute	10	3
xsd:complexType	15	2
xsd:element	23	74
Total	127	

Table 11.1: Number of named XML Schema primitives in ODRL

For instance, consider looking for all constraints in a right expression that apply to how we can

[1] http://odrl.net/

access the licensed content. This would require so many XPATH queries as there are different ways to express constraints. ODRL defines 23 constraints: industry, interval, memory, network, printer, purpose, quality... This amounts to lots of source code, difficult to develop and maintain because it is very sensible to minor changes to the ODRL specification. Fortunately, there is a workaround hidden in the language definitions.

As we have said, there is the language syntax but also some semantics. The *substitutionGroup* relations among elements and the extension/restriction base ones among *complexTypes* encode generalisation hierarchies that carry some lightweight taxonomy-like semantics. For instance, all constraints in ODRL are defined as XML elements substituting the "o-ex:constraintElement". The difficulty is that although XML Schemas provide this information, it remains hidden when working with instance documents of this XML Schemas.

Moreover, there are more complex semantics encoded in the textual definitions of the Rights Data Dictionary. They are needed each time a programmer is developing an ODRL application and thus they must be "manually" interpreted repeatedly. The idea is to make the ODRL semantics explicit in order to exploit ODRL hidden semantics and to attach more complex semantics to it. All these would facilitate ODRL applications implementation.

This objective can be accomplished using Semantic Web ontologies. OWL is used as the tool to formalise ODRL semantics. This formalisation will be accomplished in two phases. First, the lightweight semantics encoded in the ODRL XML Schemas will be translated to OWL ontologies that make them explicit. The XML Schema to OWL mapper described in the XML Semantics Reuse section 7.3 performs this translation, which is detailed in the ODRL XML Schemas to OWL section 11.1.2.

Then, it is time for the semantics informally written down as textual definitions. It is difficult to formalise them but even if the formalisation is incomplete, they will greatly facilitate ODRL applications development. In order to facilitate this formalisation, the copyright ontology semantics are reused. The terms defined by the ODRL OWL ontologies are connected to the semantically equivalent terms in the copyright ontology. Therefore, the copyright ontology rich semantics are "propagated" to the ODRL ontologies. This last step is detailed in the ODRL to Copyright Ontology Mapping section 11.1.4.

11.1.2 ODRL XML Schemas to OWL

The XSD2OWL mapping has been applied to both ODRL XML schemas. They define a quite flat set of hierarchies for *complexTypes* and *elements*. *ComplexTypes* are translated to OWL classes and they build up a hierarchy from their *extension* and *restriction base* relations as shown in Figure 11.1. Note that the "Range" suffixed and line-dotted classes correspond to implicit *complexTypes*, i.e. *complexTypes* defined implicitly inside *element* definitions that do not receive a name in the XML schema. The classes defined in the EX-11 schema are prefixed with "oex" and the ones from the data dictionary DD-11 with "odd".

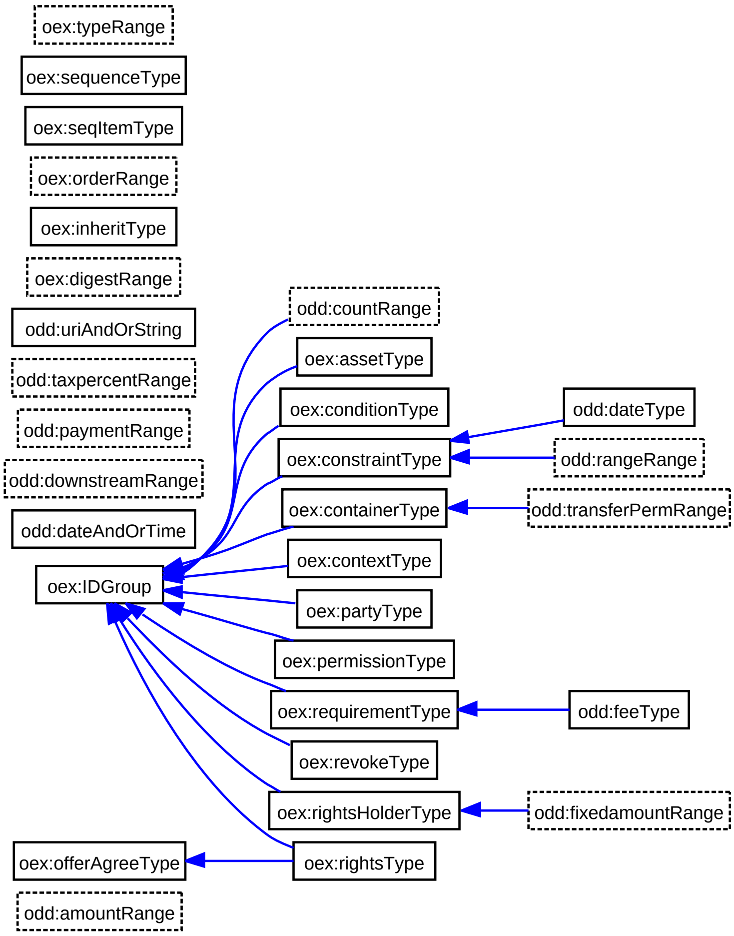

Figure 11.1: ODRL XML *complexTypes* formalised as OWL classes' hierarchies

On the other hand, *elements* and *attributes* are mapped into properties and organised hierarchically as specified by the *substitutionGroup* relation between *elements* as shown in Figure 11.2. Dark-filled properties correspond to object properties, i.e. *owl:ObjectProperty*, and light-filled ones to datatype properties, i.e. *owl:DatatypeProperty*. It has not been necessary to use *rdf:Property* because all elements have either *simpleType* or *complexType* values but not both.

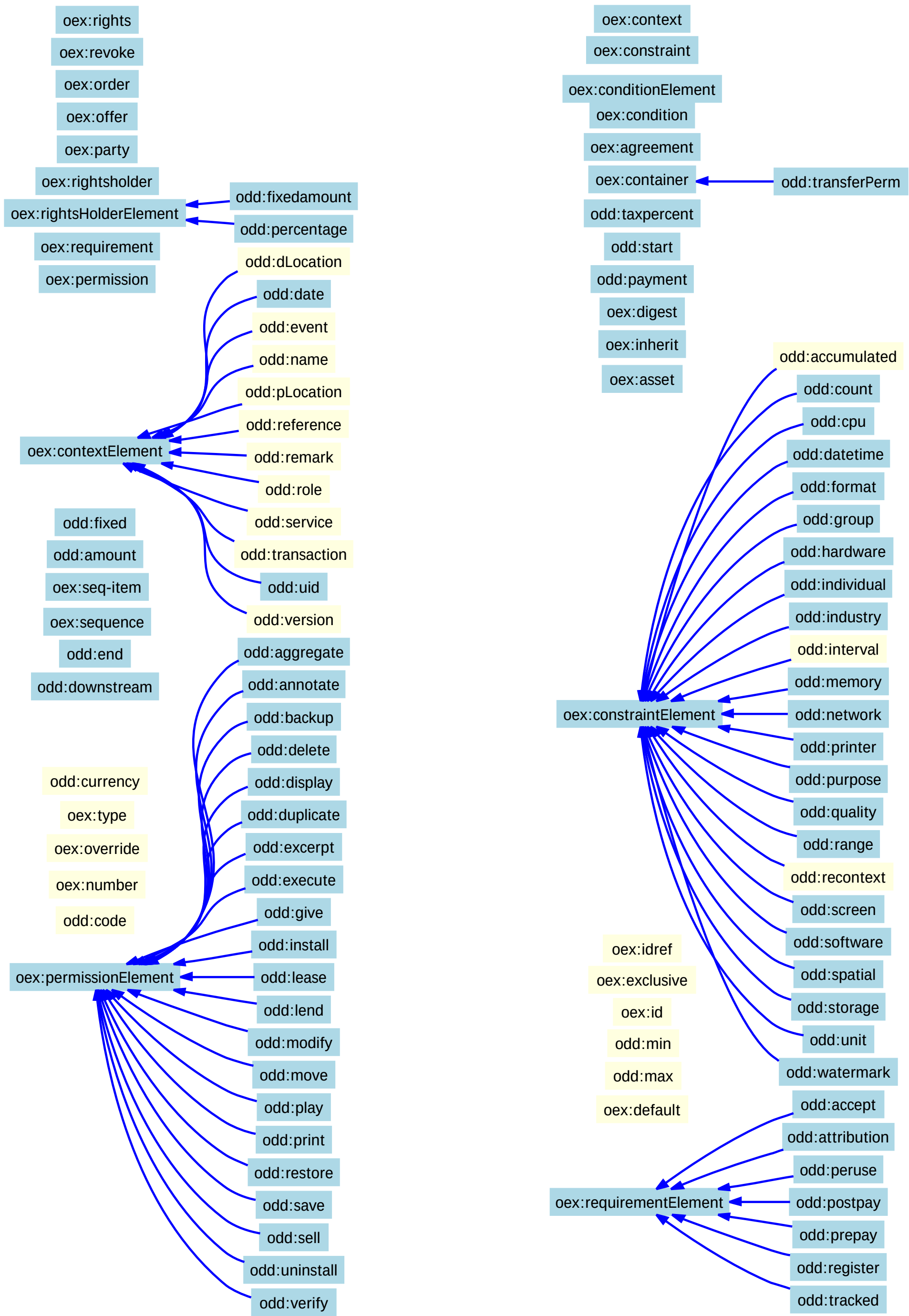

Figure 11.2: ODRL XML *elements* and *attributes* formalised as OWL properties hierarchies

The result of the XML Schema to OWL mapping is a complete OWL ontology for the ODRL language that makes the implicit semantics explicit. The OWL files for the ODRL Ontology are available at the ODRLOntos website[2]. The simplest benefit of the ODRL Ontology is that it can be used to perform semantic queries that take into account the hierarchies of *elements* and *complexTypes*. This and other advantages of ODRL semantics formalisations are detailed in the ODRL Ontology Benefits section 11.1.5.

11.1.3 ODRL XML to RDF

Applications usually operate over ODRL instances, i.e. XML documents instantiating the ORDL XML schemas. Therefore, in order to take profit from the just formalised semantics, it is necessary to map the XML instances to the semantic enriched form, i.e. to RDF metadata that instantiates the OWL ontologies just created.

The XML2RDF mapping described in the XML to RDF section 7.3.2 resolves this. It receives the XML metadata for ODRL rights expressions and produces the RDF graph that models the corresponding XML tree. As it has been shown, the RDF graph is enriched with the XML Schema hidden semantics. Now, Semantic Web tools can easily put the ODRL XML Schemas semantics into practice.

Figure 11.3 shows an example of RDF graph produced from an ODRL XML license example taken from the ODRL 1.1 specification [64]. As it can be seen, the graph models the original XML tree. The properties in the graph mimic the elements that compose the XML instance metadata. The graph is enriched with the corresponding types for the subject and objects as specified in the ODRL OWL ontology generated from the ODRL XML Schemas.

11.1.4 ODRL to Copyright Ontology Mapping

The first step of ODRL semantics formalisation provides the lightweight semantics implicit in ODRL XML Schemas. Moreover, it provides the anchor points where we are going to attach the more detailed semantics formalised from the textual definitions of the ODRL Data Dictionary. The detailed semantics are written down as text so, in order to extract them, we would need natural language processing (NLP) methods. However, NLP techniques are not advanced enough to extract the intended semantics from the short descriptions of the Data Dictionary.

Therefore, it is necessary to take a different approach. An accurate reading of the definitions

[2]http://dmag.upf.edu/ontologies/odrlontos

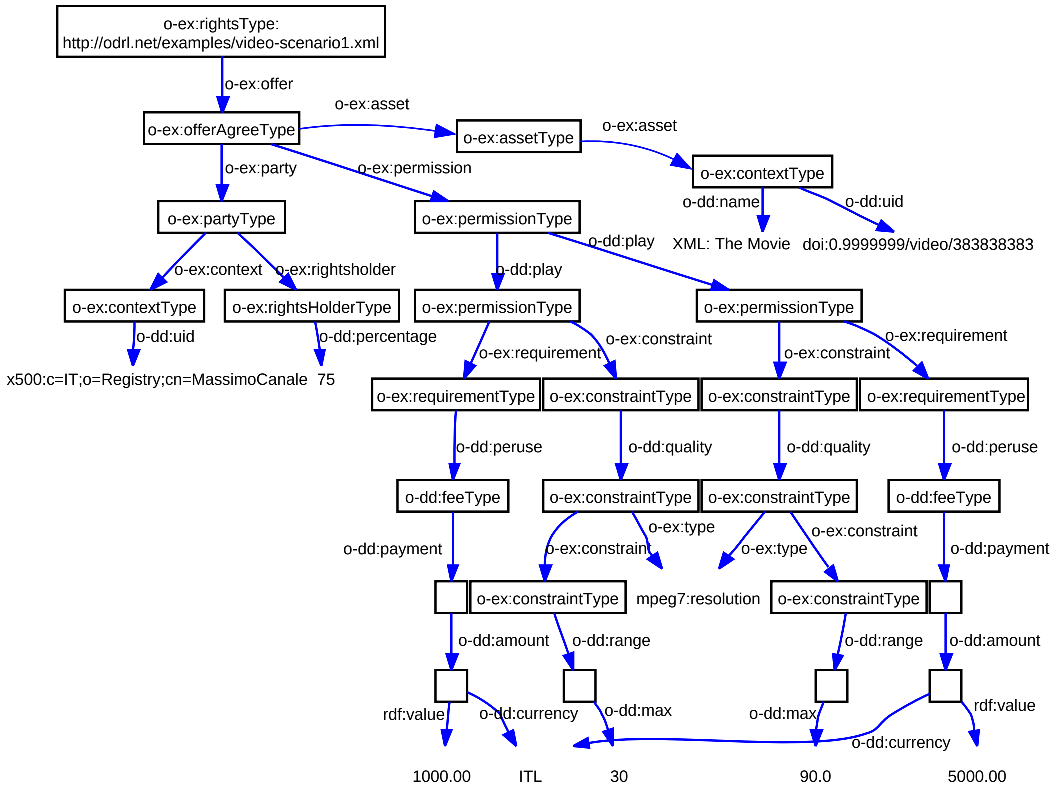

Figure 11.3: RDF mapping of an ODRL XML license example

together with the whole ODRL specification has been done, i.e. automatic means have not been used. This reading is intended to interpret ODRL semantics in the framework of the Copyright Ontology contributed in the Conceptualisation chapter 9.

The Copyright Ontology is also a OWL web ontology that provides a general semantic framework for the copyright domain. The concepts defined in the Copyright ontology are related to the concepts defined in the ODRL Ontology in order to help formalising the ODRL semantics as interpreted from the ODRL specification.

First of all, in order to facilitate mappings, some changes are introduced in the ODRL ontologies that were automatically generated from the ODRL XML Schemas. As it has been shown in the ODRL classes Figure 11.1 and ODRL properties Figure 11.2, properties, and correspondingly the ODRL XML schema *elements*, are more richly structured than classes, and consequently the ODRL XML schemas *complexTypes*.

The common situation for ontologies is the reverse one. Classes use to have richer hierarchical structure than classes and this is the case for the Copyright Ontology. Therefore, in order to facilitate mappings, the ODRL classes' hierarchy is enriched. No supplementary knowledge has been introduced. The objective is simply to replicate the properties hierarchy structure in the classes' hierarchy.

The current lack of structure is because ODRL does not define more specific *complexTypes* for "requirementType", "permissionType" and "constraintType", since they are not needed while working with XML. On the other hand, the corresponding *elements*, i.e. "requierementElement", "permissionElement" and "constraintElement", have more specific elements that appear as their subproperties in the OWL ontology, i.e. play, software, prepay, etc.

Therefore, in order to replicate structure, we introduce a new class for each one of these properties and define the class as a subclass of the corresponding existing class. For instance, the "playType" class is introduced, corresponding to the "play" property, and it is defined as subclass of "permissionType". The same is done for all the subproperties of "requierementElement", "permissionElement" and "constraintElement". The same applies for "offer" and "agree", both related to the "offerAgreeType" *complexType*. The corresponding "offerType" and "agreeType" are introduced.

As the last preparatory step, we have also reintroduced in the ODRL ontologies all the abstract elements defined in the ODRL specification but not present in the XML Schemas. Consequenly, as detailed previously, we have also introduced the corresponding classes in order to replicate the new properties in the classes hierarchy. They are "use", "reuse", "transfer" and "asset management" as "permissionElement" subproperties; "interaction", "fee" and "usage" as "requirementElement" subproperties; "user", "device", "bounds", "aspect", "target", "temporal" and "rights" as "constraintElement" subproperties.

Thanks to the previous preparatory step, the ODRL Ontology is easier to map to the Copyright Ontology. The mapping has not been implemented completely. What is detailed in this work are the principles and techniques that allow implementing them mechanically. Just some of them have already been implemented as examples.

The integration is performed using two techniques. First, for simple cases, it is possible to connect ontology concepts using OWL primitives for concept inclusion and equivalence, e.g. *subClassOf*, *subPropertyOf*, *equivalentClass*, *equivalentProperty*, *sameIndividualAs*, etc.

These are some simple mapping examples; "o-ex" prefix refers to concepts generated directly from ODRL-EX XML schema, "o-dd" for ODRL-DD XML schema, "o-ont" for the extensions generated during the previous preparatory steps and "co" for concepts in the Copyright Ontology:

- o-ex:permissionType -subClassOf→ ipro:Verb
- oddo:usageType -subClassOf→ ipro:Use
- oddo:offerType -subClassOf→ ipro:Offer
- oddo:transferType -subClassOf→ ipro:Transfer
- o-dd:individual -subPropertyOf→ ipro:agent
- o-ex:asset -subPropertyOf→ ipro:essence
- o-dd:uid -equivalentProperty→ rdf:ID
- o-dd:name -equivalentProperty→ rdf:label
- etc.

However, the previous technique is only possible when we are mapping one concept from an ontology to one concept in the other ontology. When the conditions for the mapping are more complex, the solution is to use the semantic rules described in the Semantic Web Rules section 4.8.6. Rules are particularly useful when the mapping must cope with a difference in the manner the concepts are structured in the mapped ontologies.

For instace, the ODRL context element is not used in the Copyright Ontology. Web ontologies use the RDF identifier (rdf:ID) instead of the ORDL one (o-dd:uid) and RDF identifiers are directly attached to the concept they identify. In ODRL words, this means that the identifier is a direct attribute of the asset. The same applies to the rest of the context model elements.

Therefore, the context element must be removed when mapping an ODRL instance to the Copyright Ontology. However, it is easier to convert the context of a contextualised type because it has all this information directly attached, while the contextualised type is empty. For instance, a contextualised description of an offer asset shown in Figure 11.4 is transformed using the previous simple mappings in conjunction with the mapping rule in Table 11.2 to the Copyright Ontology-aware description shown in Figure 11.5.

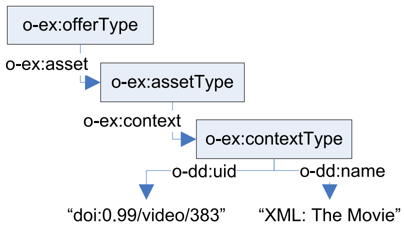

Figure 11.4: ODRL example mapped to RDF

o-ex:asset(?x,?y) ∧ o-ex:assetType(?y) ∧ o-ex:context(?y,?z) → ipro:Creation(?z) ∧ o-ex:asset(?x,?z)

Table 11.2: ODRL context to Copyright Ontology mapping rule

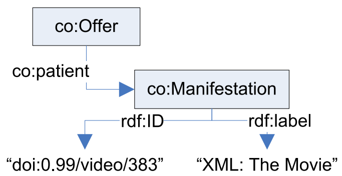

Figure 11.5: Copyright Ontoloy-aware graph resulting from Table 11.2 mapping

The previous mappings exemplify the principles that would guide a complete mapping from the ODRL Ontology to the Copyright Ontology. It remains future work as it is detailed in the Future Work chapter 13. Now it is time to show the benefits of ODRL Ontology and its mapping to the Copyright Ontology.

11.1.5 ODRL Ontology Benefits

The more direct benefit of the formal semantics provided by the ODRL Ontology is that it is possible to perform semantic queries, which are more powerful than syntax-based ones like XPath. In order to exemplify, we will retake the introduction problem about a query for retrieving the constraints affecting an ODRL rights expression. When we are working with the XML version, we need 23 different XPaths in order to retrieve all possible kinds of constraints.

This is necessary because, although XML Schemas capture some semantics of the domain they model, XML tools are based on syntax. The captured semantics remain implicit from XML processing tools point of view. Therefore, when an XQuery searches for a constraint, the XQuery processor has no way to know that there are other constraints, which can appear in its place. In other words, they are more concrete segment types.

With the RDF version connected to the ODRL ontologies, a semantic query for the super-property of all constraints, i.e. *o-ex:constraintElement*, will be automatically propagated in order to retrieve all the particular constraints defined as its *substitutionGroup* in the XML schema, and consequently defined as its subproperties in the OWL ontology. The resulting hierarchy can be seen in the ODRL properties hierarchy shown in Figure 11.2.

Semantic queries reduce the amount of source code required to develop ODRL tools and make it very easy to maintain because existing generic semantic queries can cope with changes in more specific parts of the ODRL specification without being affected. For instance, new kinds of constraints might be introduced without disturbing an previous application that implements a semantic query for *oex:ConstraintElement*.

The new constraints will be defined as *substitutionGroup* of *oex:ConstraintElement* and thus they will be mapped to subproperties of the corresponding property. The ODRL tools is always feed with the last version of the ODRL Ontology, which does not require any implementation change, so it will automatically retrieve the new constraints as the query is propagated to all the existing subproperties.

Another point is the ODRL mapping to the Copyright Ontology that makes a substantial part of the more complex part of the ODRL semantics formal. This might reduce ambiguities, or at least highlight possible ambiguous points. Moreover, there are new application development facilities. In addition to the semantic queries benefits shown before, other semantics-enabled tools can be used. One of the most promising tools is Description Logics (DL) [87].

OWL is based on DL so it can be directly fed into DL classifiers. Classifiers are specialised logic reasoners that guarantee computable results. DL classifiers are used with the Copyright Ontology in order to check copyright uses against the usage patterns specified in copyright agreements or offers as detailed in the Implementation section 10. This facilitates checking if a particular use is allowed in the context of a set of licenses or finding an offer that enables it, once an agreement is reached.

DL classifiers can be directly reused so there is no need to develop ad-hoc applications to perform this function. Moreover, as they are completely OWL semantics aware, the Copyright to ODRL ontologies mappings enables their use in order to check uses against ODRL licenses, even if they are in XML form. XML ODRL licenses can be mapped to RDF using XML2RDF and then, through mappings, are connected to the Copyright Ontology semantic framework.

11.1.6 Conclusions

As it has been shown, the Semantic Web approach to ODRL semantics formalisation has started to give its fruits. Even the first step of semantics formalisation, during which the implicit semantics of ODRL XML Schemas have been formalised, has proved very useful simply by making semantic queries possible.

The second step, during which more semantics are being defined, is showing promising results and it can greatly enlarge semantic benefits for ODRL applications implementation. Moreover, it has also allowed validating and enriching the Copyright Ontology. It has been possible to find an anchor point where ODRL Ontology concepts can be related to Copyright Ontology ones. In some cases, they were to general so more specialised intermediate concepts have been added.

To conclude, it is important to remark that all this work has been done for the current version of ODRL, version 1.1. This version was intended for XML representation and this has made the connection of ODRL ontologies to the Copyright Ontology harder. For future versions of ODRL, it might be interesting to consider this possibility, which might enable a more complete formalisation using web ontologies.

11.2 MPEG-21 REL Ontology

MPEG-21 is another Digital Rights Management initiative. It is the MPEG standardisation framework for digital content management. As it has been introduced in the MPEG-21 section, MPEG's rights modelling part is divided into the Rights Expression Language (REL) and the Rights Data Dictionary (RDD). The REL part of MPEG-21 is, like ODRL, based on XML Schemas. It is even more complex than ODRL as shown in Table 11.3, where the *attributes*, *elements* and *complexTypes* are counted for the three main MPEG-21 XML Schemas. There are 330 components for 127 in ODRL.

Schema	REL-R	REL-SX	REL-MX
xsd:attribute	9	3	1
xsd:complexType	56	35	28
xsd:element	78	84	36
Total	330		

Table 11.3: Named XML Schema primitives in MPEG-21 REL

For the XML Schemas that are part of MPEG-21 REL, the XML Semantics Reuse methodology has been also applied. The XML Schema to OWL has produced one OWL ontology for each REL XML Schema. However, this is not enough to put all REL hidden semantics into practice. That was enough with ODRL because it uses XML Schemas both for the language and dictionary definitions. However, the MPEG-21 dictionary (RDD) is not a XML Schema dictionary; it is an ad-hoc ontology. This poses additional difficulties to MPEG-21 applications development. The REL and the RDD are not integrated and RDD ontology requires specialised developments because it is not written using a common ontology language.

In order to integrate the RDD with REL, the MPEG-21 RDD ontology is also translated to OWL as it is shown in the MPEG-21 RDD Ontology section 11.3. Once this is done, this ontology is connected to the semantic formalisation build up from the MPEG-21 REL XML Schemas that are detailed in the next sections. Consequently, semantic queries will also profit from the RDD ontology semantics as it is shown in the MPEG-21 RDD Ontology Benefits section 11.3.4.

11.2.1 MPEG-21 REL XML Schemas to OWL

The same XML Semantics Reuse methodology than in the case of ODRL has been applied to MPEG-21 REL. There are three XML Schemas for MPEG-21 REL so all three have been mapped to OWL. On the contrary to ODRL Ontology, the resulting MPEG-21 REL Ontology has a more rich hierarchical structure. This is due to its greater size but also due to how the three schemas that compose the MPEG-21 REL are organised. There is the REL-R schema that corresponds to the language core entities. Then, there is the REL-SX schema that corresponds to the language standard extensions and, finally, the REL-MX schema that contains the multimedia specific extensions.

These schemas are organised as progressive semantic refinements, from the core concepts to the standard and multimedia specific ones. These corresponds to a richer hierarchy of concepts as more specific concepts are defined as specialisations of the more general ones established by the more general schemas. For instance, Figure 11.6 shows the hierarchy of concepts that specialise the MPEG-21 REL "Resource" concept. As it can be seen from the concept prefixes, the "Resource" concept comes from the Core schema, so it is prefixed "r". There are other concepts also defined in the Core schema and prefixed with "r", which constitute the more general kinds of resource in MPEG-21 REL. There are also concepts prefixed with "sx" that come from the Standard Extensions schema. These are concepts that are more specific, all of them build on top of concepts that were previously defined in the Core schema. There are no resources defined in the Multimedia Extensions schema so there are not concepts prefixed with "mx".

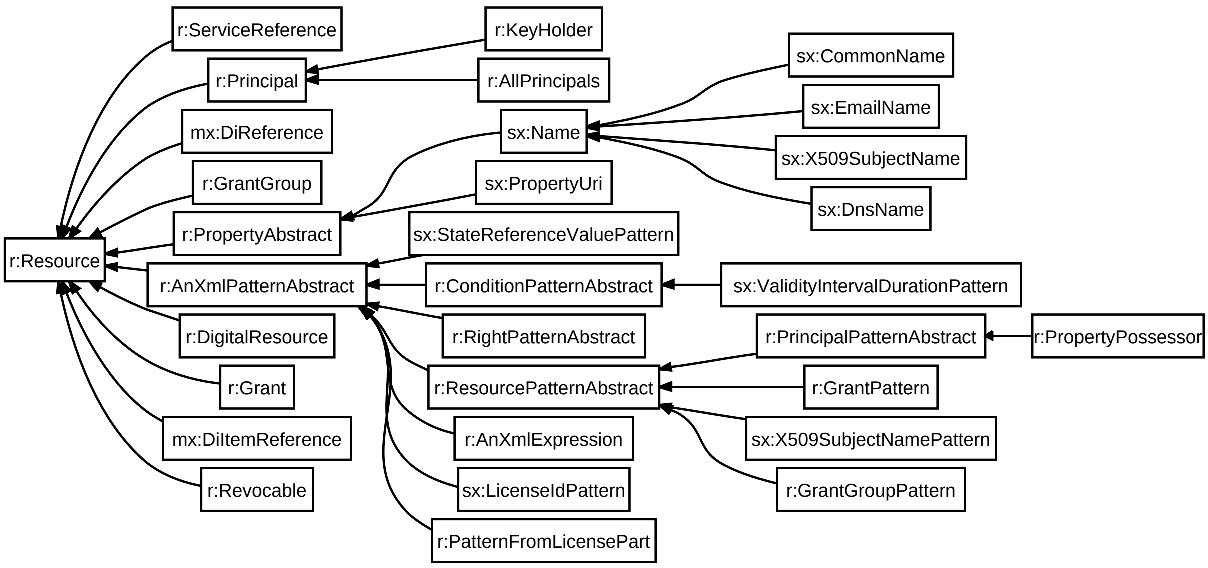

Figure 11.6: MPEG-21 REL Resource hierarchy

The result of the mapping of the three XML schemas to OWL produces a complete OWL ontology for the MPEG-21 REL that makes its implicit semantics explicit. The complete MPEG-21 REL Ontology is not shown here due to space limitations but the OWL files for the MPEG-21 REL Ontology are available at the MPEG21Ontos website[3]. The simplest benefit of the MPEG-21 REL Ontology is that it can be used to perform semantic queries that take into account the hierarchies of *elements* and *complexTypes*. However, the MPEG-21 REL was conceived to operate in conjunction with the MPEG-21 Rights Data Dictionary as it is shown in the MPEG-21 section 5.5. Therefore, the benefits of the ontological approach applied to the MPEG-21 standard are presented for both MPEG-21 REL and RDD in the RDD Ontology Benefits section 11.3.4.

[3]http://dmag.upf.edu/ontologies/mpeg21ontos

11.2.2 MPEG-21 XML to RDF

Applications usually operate over MPEG-21 REL instances, i.e. XML documents instantiating the corresponding XML schemas. Therefore, in order to take profit from the just formalised semantics, it is necessary to map the XML instances to the semantic enriched form, i.e. to RDF metadata that instantiates the OWL ontologies just created.

The XML2RDF mapping described in the XML to RDF section 7.3.2 resolves this. It receives the XML metadata for MPEG-21 REL rights expressions and produces the RDF graph that models the corresponding XML tree. As it has been shown, the RDF graph is enriched with the XML Schema hidden semantics. Now, Semantic Web tools can easily put the MPEG-21 REL XML Schemas semantics into practice.

Figure 11.7 shows an example of RDF graph produced from an MPEG-21 REL XML license example taken from the MPEG-21 REL specification [66]. As it can be seen, the graph models the original XML tree. The properties in the graph mimic the elements that compose the XML instance metadata. The graph is enriched with the corresponding types for the subject and objects as specified in the ODRL OWL ontology generated from the ODRL XML Schemas.

11.2.3 MPEG-21 REL to Copyright Ontology Mapping

In order to map the MPEG-21 REL Ontology to the Copyright Ontology, the same steps than in the case of the ODRL Ontology are necessary. However, for the MPEG-21 REL Ontology it is going to be easier because the *complexTypes* and *elements* hierarchies are identical and there is not the need to reproduce the *elements* hierarchy for the *complexTypes*. Then, in order to perform the mapping, it is necessary to implement the direct mappings based on OWL constructs for concept inclusion and equivalence, e.g. *subClassOf*, *subPropertyOf*, *equivalentClass*, *equivalentProperty*, *sameIndividualAs*, etc. Moreover, for the more complex mappings that include some structural changes in the way the information is related, it is also necessary to use the semantic rules described in the Semantic Web Rules section 4.8.6.

The ODRL Ontology mappings exemplify the principles that would guide a complete mapping from the MPEG-21 REL Ontology to the Copyright Ontology. It remains future work as detailed in the Future Work chapter 13. The benefits and conclusion of the MPEG-21 REL Ontology are presented, in conjunction with those for the MPEG-21 RDD Ontology, in the MPEG-21 RDD Ontology section 11.3.

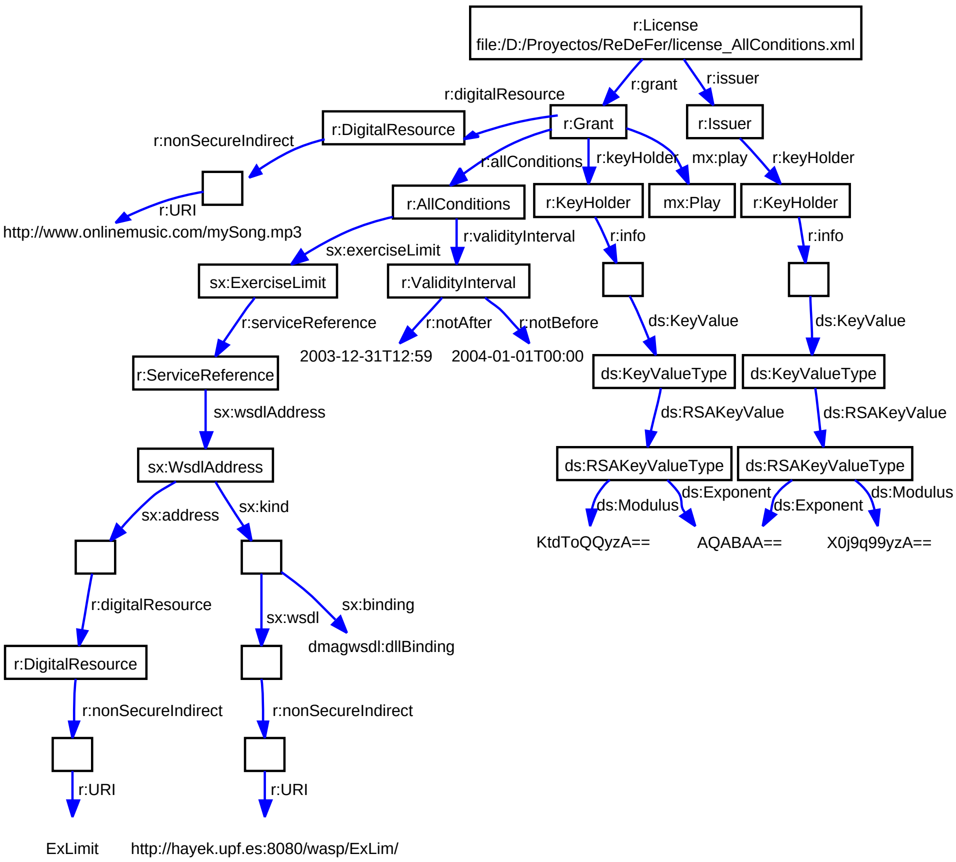

Figure 11.7: RDF mapping for a MPEG-21 REL license example

11.3 MPEG-21 RDD Ontology

The MPEG-21 RDD Ontology provides an ontological approach to the Rights Data Dictionary (RDD) part of MPEG-21. In order to build the ontology, the terms defined in the RDD specification have been modelled using OWL, trying to capture the greatest part of its semantics. On the contrary, to the ODRL and MPEG-21 ontologies, the MPEG-21 RDD is not based on XML Schemas so an ad-hoc methodology has been used to map it to OWL. The resulting ontology allows formalising a great part of the standard and simplifying its verification, consistency checking and implementation. During the RDD Ontology construction process, some integrity and consistency problems were detected, which even have led to a pair of standard corrigenda.

Additional checks were possible using Description Logic reasoning in order to test the standard consistency. Moreover, the RDD Ontology is now being used by the MPEG-21 RDD standardisation members as a tool to help verifying the standard and guide its extension. Moreover, the ontology makes automatic to integrate the RDD with other parts of MPEG-21 also mapped to OWL, e.g. the MPEG-21 REL Ontology. Finally, there are the implementation facilities provided by the ontology. They have been used to develop MPEG-21 licenses searching, validation and checking which combine the functionalities of the MPEG-21 REL and RDD ontologies. Existing ontology-enabled tools as semantic query engines or logic reasoners facilitate this.

The objective of the RDD Ontology is to translate the RDD terms descriptions from its current textual representation in the standard to a machine processable representation using the semantic web paradigm. The set of all the predefined classes and properties defined by OWL and RDF Schema are the building blocks used to model the RDD semantics.

In the RDD Specification Analysis section 11.3.1, a study of the RDD specification is presented. Then, in the RDD to Web Ontology Mappings section 11.3.2, it is shown how, first RDF Schema and afterwards OWL, can be used to capture RDD terms definitions and a great part of their semantics. RDF Schema is capable of modelling only a fraction of the RDD semantics. This fraction is augmented when the constructs introduced by OWL are also used. Therefore, two versions of the ontology can be produced. The simpler one uses RDF Schema and the more complex one uses OWL.

11.3.1 RDD Specification Analysis

The RDD Specification [67] defines a set of terms, the "words" in the vocabulary of a rights expression language. The RDD Specification is self contained so all the terms that it uses, even the relating terms, are defined in it. For each term, its description is composed by a set of attributes:

- **Headword**: the term name. It must appear in the term description.
- **Synonym**: some alternative names. It is not mandatory.
- **Definition**: a short text that defines the term.
- **MeaningType**: allowed values are: *Original*, *PartlyDerived* and *Derived*.
- **Comments**: extended textual information about the term. It is not mandatory.
- **Relationships**: this attribute lists the relationships, from a set of predefined ones, among this term an other terms. They are used to specify the term semantics from different points of views. The relations are classified in the following categories:
  - **Genealogy**: these relations give a semantic point of view similar to that from Semantic Networks [105], i.e. inheritance, relations domain and range, etc. The relations are *IsTypeOf*, *IsA*, *Is*, *IsEquivalentTo*, *IsOpposedTo*, *IsPartOf*, *IsAllowedValueOf*, *HasDomain*, *HasRange* and *IsReciprocalOf*.
  - **Types**: they are enumerated using *HasType* and its reciprocal *IsTypeOf*.
  - **Membership of Sets**: the relating term from members to sets, *IsMemberOf*.
  - **Family**: these relationships connect an *ActType* and the terms that it begets through the application of the Context Model semantics. E.g. *BegetsAgentType*.
  - **ContextView**: the group of relationships describing the attributes of a specific *ContextType* using the Context Model semantics.

11.3.2 RDD to Web Ontology Mappings

From the RDD Specification analysis two kinds of attributes can be detected. The first group is composed by those attributes with unstructured values, i.e. textual values. They can be easily mapped to predefined or new RDF properties with textual, i.e. literal, values.

The first option is to try to find predefined RDF properties that have the same meaning that the RDD term attributes that are being mapped. When this is not possible, the RDFS constructs will be used to define new RDF properties to which the corresponding attributes will be mapped. These properties are defined in the RDD Ontology namespace, "rddo".

The mappings of this kind are shown in Table 11.4. Note that the Dublin Core [24] RDF Schema is also reused in the RDD Ontology. The Dublin Core (DC) metadata element set is a standard for cross-domain information resource description. The DC RDF Schema implements the Dublin Core standard.

RDD Attribute	RDF Property	Kind of RDF property
Headword	rdf:ID	Predefined in RDF
Synonym	rddo:synonym	New property defined in the RDD Ontology
Definition	dc:description	Predefined in Dublin Core RDFS
MeaningType	rddo:meaningType	New property defined in the RDD Ontology
Comments	rdfs:comment	Predefined in RDFS Schema

Table 11.4: Mappings for the RDD attributes with text value

The other kind of attribute is the *Relationships* one. Its value is not textual. Firstly, it is categorised into five groups: *Genealogy*, *Family*, *ContextView*, *Types* and *Membership of Sets*. Each of these groups is composed by a set of relations that can be used to describe a term related to other terms in the RDD specification.

As it has been shown in the previous section, these groups of relationships take different semantic points of view. The *Genealogy*, *Types* and *Membership of Sets* groups comprise relationships with semantics almost equivalent to RDF Schema and OWL ones. The semantic equivalences have been deduced from RDD, RDF Schema and OWL specifications.

The relations in this groups that can be mapped to RDF/S are presented in Table 11.5. There is also a short description and the equivalent RDF property used to map them in the RDD Ontology. Only the RDD relations with an equivalent property in RDF Schema are mapped at this level, i.e. *IsTypeOf*, *IsA*, *HasDomain* and *HasRange*. The other relations have associated semantics that do not have equivalence in RDF Schema. Therefore, if the mapping is restricted to the possibilities provided by RDF Schema, then we get an incomplete ontology, i.e. it does not capture all the available semantics of RDD. However, on top of RDF Schema, more advanced initiatives like OWL have been developed.

Using OWL ontology building blocks, some of the previously unmapped RDD relations can be mapped to the RDD ontology. In Table 11.6 they are presented together with a short description and the equivalent OWL property used to map them in the RDD Ontology. With OWL, almost all relationships can be mapped.

Only *Is* and *IsPartOf* relations do not have equivalents in OWL. Therefore, new properties in the RDD Ontology namespace have been created to map them. Another alternative is to reuse other ontologies, as it has been done with Dublin Core. In this case, mereological (*IsPartOf*) and quality (*Is*) notions are needed. For instance, they can be reused from the DOLCE [41] foundational ontology. For *IsPartOf* the equivalent is "dolce:part-of" and for *Is* it is "dolce:has-quality".

RDD relation	Short description	RDF
IsTypeOf	Builds the hierarchy of term types	rdfs:subClassOf rdfs:subPropertyOf
IsA	Relates an instance term to its type	rdf:type
HasDomain	Defines the source term type for relations	rdf:domain
HasRange	Defines the target term type for relations	rdf:range
IsMemberOf	The RelatingTerm from Member to Set	rdfs:member

Table 11.5: Mappings for relationships in the *Genealogy*,*Types* and *Membership of Sets* groups to RDF

RDD relation	Short description	OWL
Is	Relates resources to ascribed qualities	rddo:hasQuality
IsEquivalentTo	Relates two equivalent terms	owl:equivalentClass owl:equivalentProperty owl:sameIndividualAs
IsOpposedTo	Relates two opposite terms	owl:complementOf
IsPartOf	Relates a terms that is part of another term	rddo:isPartOf
IsAllowedValueOf	Relates allowed values to a type term	Inverse of owl:oneOf
HasType	The RelatingTerm from Archetype to Type	Inverse of rdfs:subClassOf rdfs:subPropertyOf
IsReciprocalOf	For relation terms defines the relation term that captures the inverse relation	owl:inverseOf

Table 11.6: Mappings for relationships in the *Genealogy*, *Types* and *Membership of Sets* groups to OWL

For the rest of the relationship groups, a part from *Genealogy*, there is no equivalent relations in the RDF Schema plus OWL domains. This is because these relationships are based on different kinds of semantics than those used in RDF Schema and OWL. Therefore, the approach is to map them to new properties in the "rddo" namespace.

To conclude the mappings, it is also necessary to map RDD terms to Web ontology concepts. The previous mappings only cover the attributes that relate them. This has been postponed until now because Web ontology languages discern the RDD terms into three kinds: classes, properties and instances. The distinction is not made in RDD but it can be deduced from the term attributes.

If the term *Relationships* attribute includes *HasDomain* or *HasRange* relationships, it is clear that this terms must be mapped to a rdf:Property. This is a necessary and sufficient condition because all terms referring to relations have at least one of these relationships.

Otherwise, the term is a class or an instance. It will be mapped to rdfs:Class if it has a *IsTypeOf* relationship or if there is no *IsA* relationship. If there is an *IsA* relationship but not *IsTypeOf* relationship, then it will be mapped to an instance, i.e. rdf:Description. It can be noted that it is possible to have a term that has both *IsTypeOf* and *IsA* relationships that is mapped to rdfs:Class. Therefore, as specified in the OWL reference [23], the concrete OWL ontology produced is an OWL Full one.

11.3.3 Implementation

The RDD to RDF Schema and OWL mappings that have been established before have been implemented by the RDD Ontology Parser [44]. It is a Java implementation of these mappings using Java regular expressions[4]. Regular expressions are used to define patterns that detect the RDD part of the mappings. When patterns match, the corresponding RDF is generated in order to build the RDD Ontology.

Finally, once attributes have been mapped, they are used to discern the processed term as an rdfs:Class, a rdf:Property or an instance, rdf:Description. The input of the the RDD OntologyParser is a plain text version of "Table 3 - Standardized Terms" of the RDD standard [67]. The output constitutes the the RDD Ontology Web ontology available at the MPEG21Ontos website[5]. For the other relationships, a direct mapping to a new property with the same name in "rddo" namespace is implemented.

However, these relationships do not remain isolated in the resulting ontology. As all RDD terms are defined using RDD, relating terms are defined using relationships in the *Genealogy* group. Therefore, the RDD Ontology includes information about domain and range restrictions, relationships hierarchical organisation, etc.

11.3.4 RDD Ontology Benefits

The benefits of the MPEG-21 RDD Ontology are evaluated in conjunction with the MPEG-21 REL Ontology. First, it has been possible to use the RDD Ontology to check the integrity and consistency of the ISO/IEC MPEG-21 RDD standard and to amend some of the problems detected. It has been easy to integrate MPEG-21 REL and RDD, as intended from the original MPEG-21 plans but difficult to the different approaches they have taken. The integration has been easy as soon as both were in OWL ontology form. Finally, as in the case of the ODRL Ontology, it has been possible to

[4] http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/package-summary.html
[5] http://dmag.upf.edu/ontologies/mpeg21ontos

employ semantic queries and advanced reasoning tools in order to facilitate the implementation of MPEG-21 REL and RDD applications.

Checking RDD with the RDD Ontology

During the ontology development, ontology tools facilitated the detection of integrity and consistency problems in RDD. There were many references to undefined references and inconsistencies between different parts of the standard. Some of these initial problems were communicated to the MPEG-21 RDD working group and the RDD Ontology development process led to an initial revision [5] of the then recently published RDD ISO/IEC standard [67].

First, there were some inconsistencies between the textual RDD terms definitions and a figure showing the hierarchy tree of RDD act types. These inconsistencies were detected by comparing the figure included in the standard with a drawing of the Act hierarchy generated automatically from the RDD Ontology using the Protégé[6] ontology editor and the OntoViz[7] ontology visualisation plug-in.

However, the more important problems were related to the integrity issues of the standard. Some of the relationships and terms that were used in the terms definitions were not defined in it. Consequently, they have been added to the RDD Ontology, e.g. *HasCoChangedResource*, *icoInteractor*, *IsInteractorInContext*, etc. The integrity checks were performed with the help of the OWL validator vOWLidator[8].

Another testing facility once mapped to an OWL ontology is the consistency check provided by Description Logic (DL) [87] reasoners. OWL is a Description Logic so DL reasoners can be directly used in order to reason with OWL ontologies. The only limitation is that reasoners only deal with two of the three OWL sublanguages, i.e. OWL DL and OWL Lite but not OWL Full.

As it has been said, the RDD Ontology is OWL Full so we have to take away some of the mapped constraints that make it Full prior to feeding it into the DL reasoner. This has been done deactivating some of the mappings in the the RDD OntologyParser and with the further assistance of Protégé[9] combined with the Racer[10] DL reasoner. The more important feature that has been deactivated is the "IsA" to "rdf:type" mapping in order to avoid OWL Classes or Properties that are instances of other classes.

The interesting thing has been that, after making the RDD Ontology an OWL DL ontology,

[6] http://protege.stanford.edu
[7] http://protege.stanford.edu/plugins/ontoviz/ontoviz.html
[8] http://projects.semwebcentral.org/projects/vowlidator
[9] http://protege.stanford.edu
[10] http://www.racer-systems.com

we have detected 320 inconsistencies in it. All of them are due to inconsistencies between the classes and properties hierarchies. The consequence is that many property domains and ranges are inconsistent with the domains and ranges of the corresponding superproperties. For instance, the property *IsAgentActingOn* has domain *Agent*. The direct superproperty *IsRelativeOf* has domain *Relative* but *Relative* is not a superclass of *Agent* so there is an inconsistency in the *IsAgentActingOn* domain.

These results of our ontological analysis of RDD have been submitted to the MPEG standardisation group [45] and its discussion has started a process to revise the standard in order to fix these problems, which will lead to a new MPEG-21 RDD standard corrigendum.

MPEG-21 REL and RDD Ontologies Integration

The rights statements representation part of MPEG-21 is composed of the RDD, which defines the terms as it has been shown, but it also includes the Rights Expression Language (REL). The easiest way of explaining this is through a simile: the RDD provides the definition of the words while the REL provides a language to put these words together in order to build statements. However, it is difficult to put the MPEG-21 REL and RDD together into practice.

While the RDD is defined as an ontology, although a non-formal ontology language is used, REL is defined on the basis of a set of XML Schemas. This makes the integration between them very tricky. Our approach has been to take profit from the integration facilities provided by web ontologies. The REL XML Schemas have been also mapped to OWL and then easily integrated with the RDD Ontology using the OWL semantic relations for equivalence and inclusion: *subClassOf*, *subPropertyOf*, *equivalentClass*, *equivalentProperty*, *sameIndividualAs*, etc. In order to map the XML Schemas to OWL and XML instances to RDF, the XSD2OOOOWL and XML2RDF mappings have been applied.

Semantic Query

Once the REL and the RDD were integrated, it was possible to develop ontology-enabled applications that take profit from their formal semantics. This has been used to implement MPEG-21 licenses management tools. For instance, the acts taxonomy in MPEG-21 RDD, which is partly shown in Figure 11.8, can be seamlessly integrated in order to facilitate license-checking implementation. Consider the scenario: "we want to check if our set of licenses authorises us to uninstall a licensed program".

If we use a purely syntactic approach like XPath over MPEG-21 XML licenses, there must be a

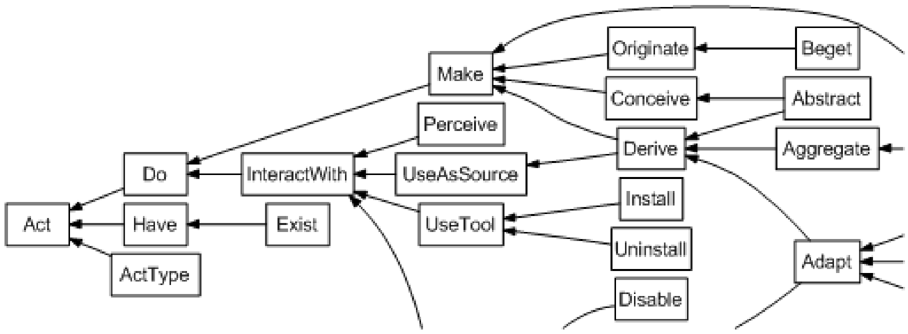

Figure 11.8: Part of the acts taxonomy in MPEG-21 RDD

path to look for licenses that grant the *uninstall* act, e.g. "//r:license/r:grant/mx:uninstall". Moreover, as it is shown in the taxonomy, the *usetool* act is a generalisation of the *uninstall* act. Therefore, we must also check for licenses that grant us *usetool*, e.g "//r:license/r:grant/mx:uninstall". In addition, successively, we should check for *interactwith*, *do* and *act*. All this must be done programmatically, the XPath queries are generated after we check the RDD ontology.

However, if we use semantic queries, the existence of a license that grants any of the acts that generalise *uninstall* implies that the license also states that the *uninstall* act is also granted. This is so because, by inference, the presence of the fact that relates the license to the granted act implies all the facts that relate the license to all the acts that specialise this act.

Therefore, it would suffice to check the semantic query "//r:license/r:grant/mx:uninstall". If any of the more general acts were granted, it would match. For instance, the XML fragment "/r:license/r:grant/dd:usetool" implies the fragments "/r:license/r:grant/dd:install" and "/r:license/r:grant/dc

Usage against License Checking Using DL Reasoners

There are other application development facilities more sophisticated than the semantic queries benefits shown before. One of the most promising tools is Description Logics (DL) [87]. OWL is based on DL so it can be directly fed into DL classifiers. Classifiers are specialised logic reasoners that guarantee computable results. DL classifiers are used with the MPEG-21 REL and RDD ontologies in order to check content uses against the usage patterns specified in copyright agreements or offers as detailed in the Implementation section 10. This facilitates checking if a particular use is allowed in the context of a set of licenses or finding an offer that enables it, once an agreement is reached.

DL classifiers can be directly reused so there is no need to develop ad-hoc applications to perform this function. Moreover, as they are completely OWL semantics aware, the Copyright to MPEG-21 ontologies mappings enables their use in order to check uses against MPEG-21 licenses, even if they are in XML form. XML MPEG-21 REL licenses can be mapped to RDF using XML2RDF and then, through mappings, are connected to the Copyright Ontology semantic framework.

11.3.5 Conclusions

The benefit of the MPEG-21 RDD and REL ontologies over other initiatives is that it is based on applying an ontological approach. This is done by modelling the RDD standard using ontologies. Ontologies allow that a greater part of the standard is formalised and thus more easily available for implementation, verification, consistency checking, etc.

The MPEG-21 RDD Ontology demonstrates the benefits of capturing the RDD semantics in a computer-aware formalisation. First, it has been possible to analyse the standard integrity and consistency with the support of ontology-aware tools that facilitate this issue, discovering inconsistencies that are in the process of being fixed in the standard. Then, it has been possible to integrate RDD with another MPEG-21 standard part, the Rights Expression Language (REL), in a common ontological framework. This framework facilitates the implementation of MPEG-21 tools.

We have shown our achievements using semantic query engines and Description Logic reasoners for license searching, validation and checking. The ontological approach has also made possible the development of advanced Digital Rights Management systems that integrate these tools in order to build semantic information systems, as the one presented in the Semantic DRM Systems section 11.4, and intelligent agents for assisted rights negotiation as detailed in the Negotiation Support section 11.4.3.

The objective now is to take profit from the abstraction and integration facilities of formal ontologies in order to cope with the RDD standard problems. First of all, the RDD Ontology is being used in order to extend RDD capabilities in a consistent and more informed way. Some communities might find that there are some unsatisfied requirements in the current RDD. This is completely normal as it is impossible to cope with all the requirements of communities as big as the ones that might be interested in the MPEG-21 standard.

The MPEG-21 RDD standard specifies mechanisms for standard extension. However, it is difficult to put these mechanisms into practise. The size of the standard makes it very complex for people outside the standardisation process to manipulate and extend it in order to satisfy their particular needs. This is why we have started to use the RDD Ontology as an assistance mechanism for RDD testing of new requirements. The RDD Ontology is used together with ontology rendering tools in order to navigate the RDD hierarchy of concepts, detect the part of it where the new concept might be situated and even produce a graphical drawing of it.

Another future line is to exploit the integration possibilities of OWL in order to connect the RDD Ontology with more general ontologies, as the Copyright Ontology, or rights data dictionaries of other rights expression languages like ODRL. The objective here is to build an ontology-based framework that allows integrating these initiatives, making them interoperable and enrich them with the possibilities offered by formal ontologies. This might lead to levels of interoperability that allow combining different RELs and RDDs in a totally uncoupled way.

11.4 Semantic DRM System

Digital Rights Management (DRM) is a complex domain. The DRM field is structured by evolving regulations, practises, business models, etc. Therefore, DRM Systems (DRMS) are very difficult to develop and maintain. The ontological approach to DRM contributed by this work helps dealing with DRM complexities as it has been already shown in the previous validation sections. However, these ontologies need to be put into practice in order to show their benefits.
The NewMARS DRMS [46], formerly called MARS [43], is semantics-enabled DRM system that puts the ontology-based approach to DRM into practice and validates its usefulness. A knowledge-oriented approach has been chosen in order to make this development capable of dealing with this complicated domain. This requirement and the objective of easy Web integration have made the Semantic Web technologies the best choice.
Semantic metadata is associated to copyright governed content using URIs and it is structured using web ontologies. There are descriptive, rights and e-commerce ontologies for the different views on copyright content. Semantic enabled metadata is then used to facilitate content providers to publish content offers and customers to find and automatically negotiate purchase conditions. All this is performed under the copyright governing premises.
All NewMARS modules are interrelated using the ontologies shared semantics. This has allowed developing very flexible project infrastructures that facilitates easy adaptation to new copyright e-commerce scenarios.

11.4.1 Introduction

This Semantic DRMS tries to make a new contribution to the DRM field. DRM has been strongly affected by the digital era changes. Even now, all the new situations related to copyrighted content arisen from digitalisation and the Internet has not been satisfactorily resolved.

Some of these problems are faced by current initiatives trying to solve interoperability between Digital Rights Management (DRM) systems. DRM systems started from isolated and proprietary initiatives. However, they are lately moving to a web-broad application domain due to the World Wide Web effect on the digital content market.

There are many other initiatives but, basically, all have one thing in common, they work at the syntactic level. Their approach is to make a formalisation of some XML DTDs and Schemas [123] that define a rights expression language (REL). In some cases, the semantics of these languages, the meaning of the expressions, are also provided but formalised separately as rights data dictionaries (RDD). Rights dictionaries list terms definitions in natural language, intended for human consumption and not easily automatable.

However, this kind of syntactic approaches are not solving the problem as a whole. They do not scale well in really wide and open domains like the Internet. Therefore, the interoperability problems are reappearing, as it is very difficult to establish a one-fits-all standard.

Most probably, we are not going to see a clear winner in the REL battlefield, at least in the short time range. However, automatic processing means for the huge and heterogeneous amounts of metadata produced by DRM are required. The syntax is not enough when unforeseen expressions are met. Here is where machine understandable semantics come to help metadata interpretation to achieve interoperability.

Our idea is to facilitate the automation and interoperability of DRM frameworks integrating both parts, the Rights Expression Language and the Rights Data Dictionary. As it has been shown in this work, these objectives can be accomplished using ontologies, which provide the required definitions for the REL terms in a machine-readable form. Thus, from the automatic processing point of view, a more complete vision of the application domain is available and more sophisticated processing can be carried out.

Once the ontological framework based on the Copyright Ontology has been laid out, it is just a formalisation without utility if it is not put into practice. This has been the objective of the NewMARS project: to build a DRMS that takes profit from the advantages of the ontologies formalisations, which will facilitate the implementation of copyrighted content e-commerce solutions.

11.4.2 Application Domain

In order to put NewMARS into practice, what has been done first is to analyse the DRM business model. This business model defines the environment where NewMARS will fit, the actors with which it will interact and the interaction rules. The business model we have considered is presented in the Business Model section 11.4.2.

The NewMARS Project planning has been guided by the idea to make a knowledge-guided development, from a computer point of view. This implies transferring a great amount of the development effort from the functional model to the domain knowledge model.

Consequently, the number of application functions is reduced to some basic ones in charge of message interchange among the application parts. A user actions diagram detailing actors and functions is detailed in the User Actions Analysis section 11.4.2. Therefore, the focus is placed on the semantics of these messages.

As it has been introduced before, the Copyright Ontology is used as the basis of the knowledge model. Therefore, a great part of this effort has been already done and it is reused in NewMARS. There are only some small extensions to the knowledge model derived from the practical aspects of the project. More details about this are given in the Metadata section 11.4.2.

Business Model

The e-commerce of copyright is guided by a business model that has emerged from the associated regulations framework, the commercial activity and the electronic means that have influenced it.

In order to build NewMARS upon a quite generic and flexible business model inspired by the one produced by the IMPRIMATUR project [6], the NewMARS business model identifies a set of basic roles and interactions among them. These basic roles are shown in the centre of Figure 11.9 and they constitute a generic value chain.

In parallel, some support services have been also identified. They constitute the basic services that facilitate the IPR e-commerce activity. They are shown in Figure 11.9 round the roles to which they give support along the whole value chain.

To facilitate the implementation of this model, it has been combined with a broker-based e-commerce model that has been extensively tested in previous research [40, 28, 43]. The final broker-based business model implemented in NewMARS is shown in Figure 11.10.

The broker facilitates value chain actors access the DRM e-commerce services. Moreover, in

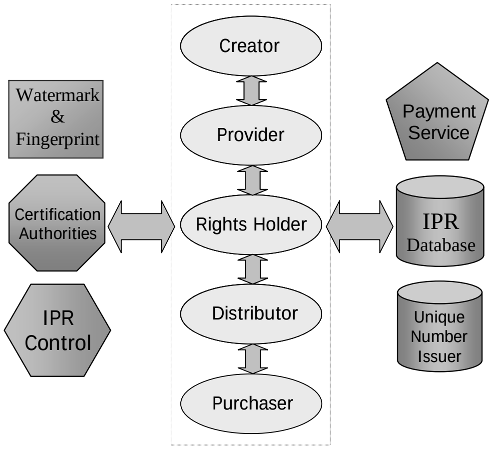

Figure 11.9: Generic DRM Business Model

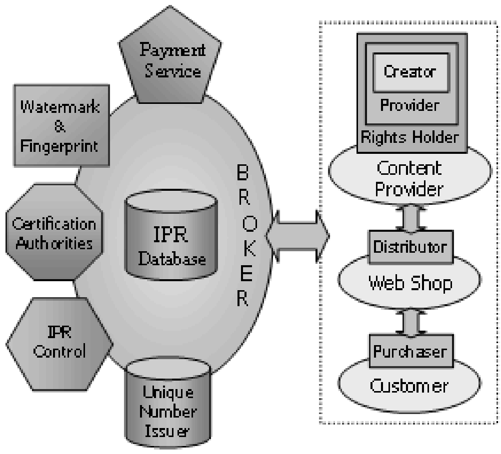

Figure 11.10: NewMARS Broker-based DRM Business Model

the NewMARS scenario, actors have been simplified to three, each one playing one or more roles: Content Provider (it can play the Creator, Provider and Rights Holder roles), Web Shop (it plays the Distributor role) and Customer (it plays the Purchaser role).

In addition to the broker, the NewMARS project is also going to implement the Distributor role through a web shop. Consequently, there will be only two external actors: Content Provider and Customer. More details are given in the user actions analysis in the next section.

Use Cases Analysis

Figure 11.11 shows the use cases that specify the relations among the external actors that have been identified and the application.

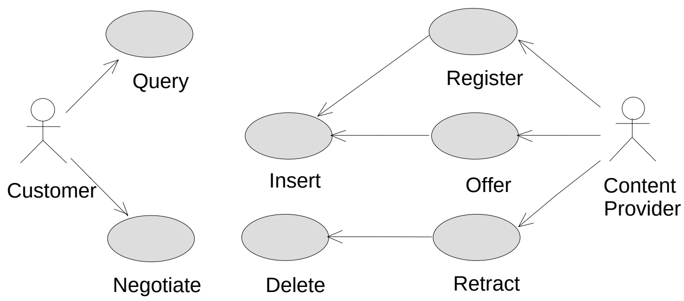

Figure 11.11: NewMARS Use Cases Diagram

The user actions are detailed below:

- **Insert**: this "internal" action is not directly accessible to external actors. Its functionality is accessed from other actions. What this action does is to store information about a resource into the NewMARS system. Due to the knowledge-oriented approach, this action can be viewed as the assertion making one.
- **Delete**: it is also "internal" but it is the counterpart of the previous action. It is responsible for un-asserting facts.
- **Register**: content providers use this to add new information about the content they manage. The information chunks are sets of assertions describing the content and their rights situation.
- **Offer**: this action is accessed by the Content Provider to add e-commerce information about a content that is offered under a given conditions.

- **Retract**: content providers can delete information chunks about content they have previously inserted in NewMARS. This includes descriptive, rights and e-commerce information.
- **Query**: customers can use this action to look for desired content. The queries submitted by the customer are matched against descriptive, rights and e-commerce information stored in NewMARS. In return, the customer receives all the registries associated to the resources that have matched the criteria.
- **Negotiate**: once e-commerce information has been retrieved, if it does not completely satisfy the customer it can be negotiated. When a satisfactory offer is achieved the customer can accept it, then it is fulfilled.

Metadata

The content information that is managed by NewMARS is modelled as metadata associated to resources. Moreover, a set of ontologies provide the required semantics. As it has been introduced before, the Copyright Ontology is used as the foundation for rights and e-commerce metadata.

However, descriptive metadata depends on the particular content that is managed. Due to project requirements, NewMARS was planned considering digital multimedia content. Therefore, ontologies about descriptive metadata for this kind of content where considered.

The MPEG-7 standard [101] was taken as the source for the descriptive ontology due to its coverage and relevance. MPEG-7 is based on XML Schema so the XML Semantics Reuse methodology was employed to generate a MPEG-7 OWL ontology [42], which was complemented with TV-Anytime[11] ontologies also generated from their XML descriptions. All these ontologies are available from the MPEG-7 Ontologies website[12]. This constitutes a big source of semantic multimedia metadata.

The previous descriptive ontology provides a quite satisfactory framework for multimedia content description. The multimedia specific aspects are complemented with the generic ones provided by Dublin Core [24]. An example of RDF metadata description in NewMARS is shown in Table 11.7.

Another key element about metadata in NewMARS is that it is expected to come from many different sources, i.e. metadata stores. Therefore, it is required that the metadata management processes implemented support this feature. However, from the outside, the users should experience an integral view of metadata so the metadata must be merged transparently.

[11] http://www.tv-anytime.org/
[12] http://dmag.upf.edu/ontologies/mpeg7ontos

```
< xml version="1.0" encoding="UTF-8" >
<!DOCTYPE rdf:RDF [
<!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
<!ENTITY dc 'http://purl.org/dc/elements/1.1/'>
<!ENTITY dctype 'http://purl.org/dc/dcmitype/'>
<!ENTITY xsd 'http://www.w3.org/2001/XMLSchema#'>
<!ENTITY mpg7 'http://dmag.upf.edu/ontologies/2005/03/Mpeg7-2001.owl#'>
<!ENTITY tva 'http://dmag.upf.edu/ontologies/2005/03/TVAnytimeMetadata.owl#'>
<!ENTITY tvac 'http://dmag.upf.edu/ontologies/2005/03/TVAnytimeContent.owl#'>
<!ENTITY tvaf 'http://dmag.upf.edu/ontologies/2005/03/TVAnytimeFormat.owl#'>
<!ENTITY ipr 'http://dmag.upf.edu/ontologies/2005/05/IPROnto.owl#'>]>
<rdf:RDF xmlns:rdf="&rdf;" xmlns:owl="&owl;" xmlns:dc="&dc;" xmlns:mpg7="&mpg7;"
xmlns:tva="&tva;" xmlns:tvac="&tvac;" xmlns:tvaf="&tvaf;" xmlns:ipr="&ipr;">
  <ipr:Work rdf:about="urn:iswc:FF-Wind_Power">
<tva:Genre rdf:resource="&tvac;1.13.7"/>
<tva:Genre rdf:resource="&tvac;1.13.9"/>
<tva:Genre rdf:resource="&tvaf;1.6"/>
<dc:title xml:lang="en">The wind power as an alternative at Catalonia</dc:title>
<dc:description xml:lang="en">The wind power </dc:description>
<dc:type rdf:resource="&dctype;MovingImage"/>
<dc:language>ca</dc:language>
<dc:date rdf:datatype="&xsd;date">2000</dc:date>
<dc:creator rdf:resource="http://www.ff.com"/>
</ipr:Work>
  <ipr:Fixation rdf:about="urn:isan:FF-Wind_Power.mpg">
<ipr:isFixationOf rdf:resource="urn:iswc:FF-Wind_Power"/>
<dc:format>video/mpeg</dc:format>
</ipr:Fixation>
  <ipr:Agree rdf:about="urn:agreement:200505111210">
<ipr:agent rdf:resource="urn:x500:CN=U26473,O=PROVIDERS,C=ES"/>
<ipr:start rdf:datatype="&xsd;date">2005-04-01</ipr:start>
<ipr:theme>
<ipr:Access>
<ipr:agent rdf:resource="urn:x500:O=USERS,C=ES"/>
<ipr:theme rdf:resource="urn:isan:FF-Wind_Power.mpg"/>
<ipr:location rdf:resource="http://providers.net/videos/FF-Wind_Power.mpg"/>
<ipr:duration rdf:datatype="&xsd;duration">PT10H</ipr:duration>
<ipr:condition>
<ipr:compensation>
<ipr:recipient rdf:resource="urn:x500:CN=U26473,O=PROVIDERS,C=ES"/>
<ipr:theme>
<ipr:CurrencyMeasure rdf:value="2">
<ipr:currencyUnit rdf:resource="&cur;EUR"/>
</ipr:CurrencyMeasure>
</ipr:theme>
</ipr:conpensation>
</ipr:Access>
</ipr:theme>
</ipr:Agree>
</rdf:RDF>
```


Table 11.7: NewMARS metadata example

In order to implement this feature, the best option is to use RDF metadata through all the NewMARS information flows. Therefore, NewMARS receives RDF metadata as input, manages it and also produces RDF metadata as output. When RDF metadata coming from different sources must be combined, the RDF graph model facilitates metadata integration that is reduced to a process of graph merging. Once integrated, the metadata graph can be serialized and sent to the output. More details about how this is implemented are shown in the Metadata Retrieval section 11.4.3.

11.4.3 Development

Once the application domain has been introduced, this section details how the application has been developed. The driving force has been knowledge orientation. This has been materialised by prioritising application modules decoupling and basing module interrelation in shared semantics.

Web technologies, and more concretely Semantic Web tools, have been chosen as the more appropriate ones considering these requirements. First of all the following technological choices have been realised:

- Message transport: SOAP [117].
- Message encoding: RDF [7].
- Message semantics: ACL [37].
- Ontology language: OWL [23].
- User interface: HTML.
- Negotiation: JADE+JESS [8, 39].

From the combination of requirements, design principles and technological choices, the architecture shown in Figure 11.12 has emerged.

The architecture defines three main blocks:

- **Broker and Storage**: this block is in charge of the main NewMARS responsibilities, i.e. all actions apart from "Negotiate". The broker offers a SOAP interface through which it interchanges SOAP messages. These messages are encoded using RDF and then structured using a web ontology that models FIPA ACL (Agent Communication Language) in order to provide message semantics. Message semantics define which messages are queries, facts assertions or facts removals. In each case, independent metadata stores are accessed for metadata retrieval, insertion or deletion. More details in the Broker and Storage section 11.4.3.

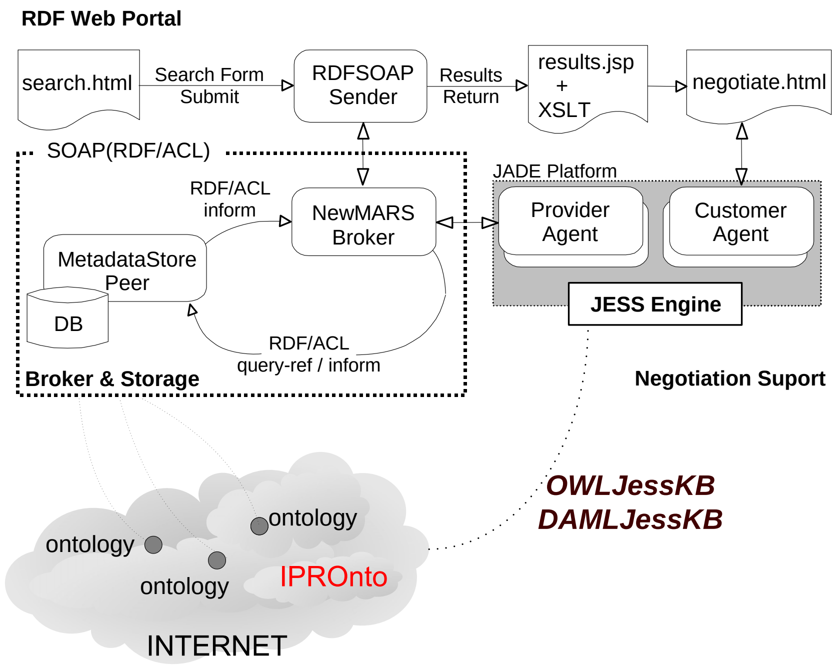

Figure 11.12: NewMARS architecture

- **Web Portal**: this block is the front-end that interacts with external users. The objective of this block is to provide an easy and common user interface, so HTML has been selected. In order to interact with the broker the RDFSOAPSender has been developed. First, RDFSOAPSender facilitates sending messages to the broker: it encapsulates HTML forms submissions as RDF/ACL messages and sends them using SOAP to the broker. Second, it manages messages responses: it processes the return messages in order to transform their RDF content to HTML that can be shown to the user. This block is detailed in section the Web Portal section 11.4.3.

- **Negotiation Support**: this block is responsible of giving service to the "Negotiate" action. The objective is to offer automatic or semiautomatic negotiation support to users. Agents technology is used to perform this. We have chosen JADE as the multi-agent platform because it provides agent technology building blocks and implements FIPA standards. Agents decision support is managed by the JESS expert system. More details are given in section Negotiation Support section 11.4.3.

Broker and Storage

As it has been introduced before, the broker block of NewMARS has a SOAP interface. However, this interface is only used for message transport. Thus, message semantics do not depend on different SOAP interface methods. Message semantics are determined by their structure and content.

The ACL Ontology[13] is used to define message structure. The structure determines what to do with message content, which can be a query or metadata like those presented in the Metadata section 11.4.2. The actions that can be taken by the broker are at last supported by the metadata store elements that allow metadata storage and retrieval.

Message Structure Message structure is based on the Agent Communication Language. ACL [37] defines a set of communicative acts that establish message intentionality, i.e. its pragmatics. ACL also defines attributes that determine message characteristics. Some of these communicative acts are used in messages sent to the broker because they correspond to the user actions it manages:

- **Insert and Delete**: this action is captured by the *inform* communicative act when a chunk of metadata is "informed" to the broker. When a reference (URI) pointing to the metadata is communicated the *inform-ref* act is used. The *inform* can be used to assert affirmative or negative facts, i.e. unassert. The broker responds with a *inform* message to communicate insertion outcome.
- **Query**: this action corresponds to the *query-ref* act. It is a query by reference, where the reference is the pattern encoded by the query sentence. There are many RDF query languages so the language attribute is used to tell the broker which one is used. The broker responds to the query with an *inform* message.
  The message semantics defined by ACL are used by the broker to route them to the appropriate metadata store peer as detailed next. The appropriate store is determined by the broker, for instance by considering the message language attribute.

Metadata Storage

The different broker actions end up with an access to the metadata storage system. As it has been shown in the architecture, it has been separated from the broker in different independent modules. Communication between the broker and the selected metadata storage peer is also performed by means of ACL structured messages.

[13] http://www.cs.umbc.edu/~yzou1/daml/acl.daml

The message communicative act tells the store peer how it has to interpret it. The content of inform messages is inserted or deleted and query-ref messages content is interpreted as query sentences.

The store peer is supported by a RDF store that is in charge of really storing the metadata or retrieving the stored metadata corresponding to the pattern determined by the query sentence. The store peers make the broker and all the application independent from the particular RDF store used. Therefore, they show the same behaviour. They receive RDF metadata as input of Insert and Delete actions.

MetadataStore Peers are not only responsible for making the NewMARS system independent from the different metadata store particularities. Moreover, they are also responsible for converting metadata query results from the common table-like result sets to RDF metadata as it is detailed next.

Metadata Retrieval As has been shown in the previous section, the Broker receives RDF metadata as input. This is a common behaviour of RDF stores so, in this case, little work has to be done.

On the other hand, as it has been stated during the application domain analysis, it is also very interesting to get RDF metadata as output from RDF stores so the whole information flow is done in RDF form. This has been justified as it facilitates the integration of metadata coming from different sources.

Moreover, if the web portal receiving the output from the NewMARS broker gets RDF metadata instead of table-like result sets, more information would be available in order to render this metadata to the user. In other words, the stored metadata semantics would not be lost in the query output and would arrive intact until the last information-processing step.

In this case there is some work to do as producing RDF metadata as query output is a very uncommon behaviour of RDF metadata stores. Query sentences are augmented by the NewMARS Broker with a special construct "graph(*sentence*, *depth*)". When this construct is sent to a store peer, it indicates that the store peer has to construct one or more RDF graphs from the resources selected with the query sentence.

This is done by retrieving RDF triples from the selected resource to the maximum depth specified. However, blank nodes are not considered when computing this depth; i.e. triples with blank node resources are always added if they are directly connected to selected resources or indirectly through a chain of blank resources.

For example, see Figure 11.13. From a query that selects the resource "urn:iswc:FF-Wind_Power", the graph construction algorithm is applied with depth equal to one. All the grey filled resources and literals and the solid line properties are retrieved. The Bag anonymous resource is ignored in order to compute depth so its members and its type are also retrieved. On the other hand, the metadata attached to the Video and Documentary types, i.e. the white filled resources and literals and the dotted line properties, are not retrieved as they are at a greater depth.

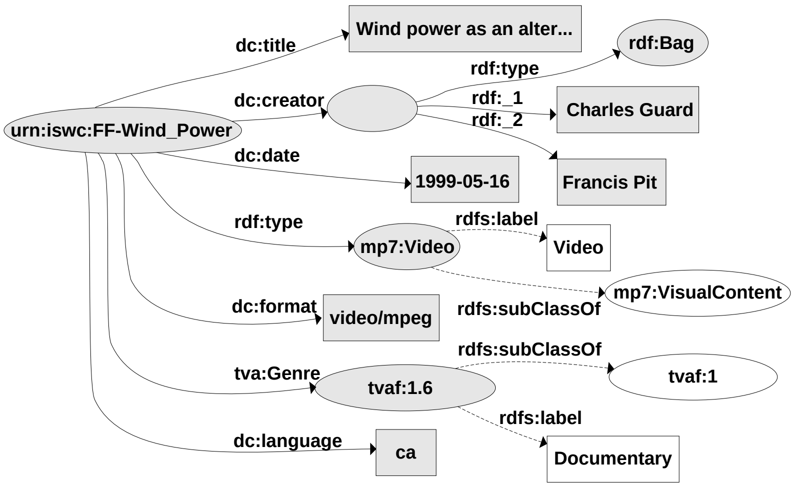

Figure 11.13: Graph construction example for metadata retrieval

Once the query response graph or graphs have been constructed, they are serialised as RDF/XML and encapsulated in the response messages. They are structured as inform messages containing the response metadata. As it has been shown, store peers allow a great independence from the concrete RDF stores used. Currently two RDF stores have been integrated: RDF Suite[14] and Sesame[15].

Web Portal

The web portal has been developed as the user interface to the NewMARS functionality. The application has been developed based on the interchange of RDF messages with SOAP transport. Therefore, the portal must have a mechanism to encapsulate user interactions as RDF/ACL messages and send them to the broker by SOAP. Moreover, the responses to user interactions are made visible to the user by translating them to HTML. The web portal functionality is detailed in the next subsections.

RDFSOAPSender This is the module responsible for the interaction between the portal and the broker. It is a servlet that receives user commands encoded as HTML form submissions. The form

[14] http://www.ics.forth.gr/proj/isst/RDF
[15] http://www.openrdf.org/

parameters are transformed into RDF triples, one for each parameter. All the triples have the same resource that identifies the current command. The properties are the parameter names and the resources their values.

The triples are serialised as RDFXML that is inserted into a new SOAP message in order to be sent, as shown in Table 11.8. The RDF content of the messages is built from the parameters received from the HTML forms through which the users interact with the system. Three basic forms can be identified: Query, Register/Offer and Retract.

```
<SOAP-ENV:Envelope xmlns: >
<SOAP-ENV:Body>
<rdf:RDF ... >...</rdf:RDF>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Table 11.8: SOAP envelope used to transport RDF/ACL messages

Query Form This form is composed by a set of fields relative to the attributes that finally will compose the RDF/ACL message that the RDFSOAPSender is going to generate. The available fields in the Query form are:

- **Sender**: the form web page URL or the identifier with which the user has identified himself in the web portal.
- **Receiver**: the broker URL where the SOAP message will be sent.
- **Reply-to**: the URL where the results will be sent in order to show them.
- **Content**: the query sentence.
- **Language**: the query language. Current RDF stores (RDF Suite and Sesame) use RQL [70] and SeRQL [69]. However, other possibilities can be easily incorporated.
- **Performative**: it indicates the message communicative act. For the query form it is fixed to the query-ref act.

Table 11.9 shows an example of RDF/ACL message built from a query form submission. It is an RQL query that retrieves metadata associated to offers that allow access to multimedia contents. The response is redirected to a web page that will format the output RDF metadata as HTML.

```
<rdf:RDF xmlns:acl="http://daml.umbc.edu/acldaml" ...>
<acl:query-ref>
<acl:sender>http://dmag.upf.edu/newmars/search.html</acl:sender>
<acl:receiver>http://dmag.upf.edu/newmars/broker</acl:receiver>
<acl:language>RQL</acl:language>
<acl:content parseType=Literal>
graph(select X,Y
from {X;Offer}permission{;Access}.patient{Y;AudioVisual})
</acl:content>
<acl:reply-to>http://dmag.upf.edu/newmars/results.jsp</acl:reply-to>
</acl:query-ref>
<rdf:RDF>
```

Table 11.9: Example of RDF/ACL message built from a query form submission

Register/Offer Form This form is used to tell the broker the IP descriptive, rights or e-commerce metadata to be inserted in the system. It is like the previous form. The only changes are performative, inform or inform-ref, and language that now is RDF/XML in order to reflex that the content is RDF metadata.

Metadata Web Rendering The result web pages use XSL style sheets [119] in order to transform the RDF metadata form response messages into HTML that can be shown by the web portal. There is a basic style sheet responsible for transforming each RDF description in the response metadata into an HTML table.

Each row corresponds to one property directly associated to the description. The first column is the property id and the second column is the property value. If the value is another resource, a sub-table is recursively inserted and the whole table construction process is repeated.

This basic XSL style sheet is then combined with particular ones that complete HTML layout in order to particularise output to the special needs required. An example of complete HTML layout of a RDF encoded offer is shown in Figure 11.14.

As has been shown in the Metadata Retrieval section 11.4.3, the metadata that is rendered is collected by building graphs from the selected resources to a given depth, commonly with depth one. In many cases this produces bunches of metadata with the relevant information for the posed query. However, sometimes it is necessary to get deeper in the graph and retrieve more metadata.

In order to facilitate metadata navigation, the XSL style sheet also produces HTML links for all the resource URLs. This links correspond to queries to the NewMARS broker for metadata about the clicked resource. Then, a pop-up window is opened showing the new metadata detail. The same XSL style sheet is applied to it so new HTML links are generated and they allow continuing

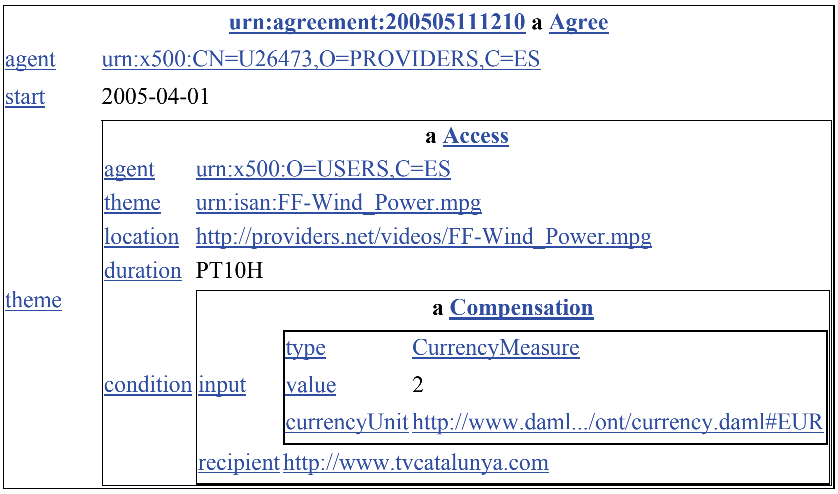

Figure 11.14: HTML render from the RDF-encoded Offer in Table 11.7

the RDF metadata browsing experience through HTML. It can be tested on-line in the NewMARS website[16].

Negotiation Support

Agents technology is used to perform negotiation. Negotiation is the last customer action. It is performed once the customer has located the desired content and the corresponding offer that is going to be negotiated. Offers can be directly accepted, rejected or negotiated.

We have chosen the JADE multiagent platform. In order to reason about facts coming from messages, JESS (Java Expert System Shell) [39] has been used because it is easy to make it work together with JADE.

If the customer wants to negotiate the offer, he can choose a personal agent that will intermediate between the customer and the content provider agent. Customer and content provider agents are JADE agents controlled by the expert system. They negotiate the license offers.

The negotiation protocol is controlled by JESS and this allows a dynamic negotiation between the agents, who make offers and counteroffers, and it allows to process licenses. There are two main phases in the negotiation and they are introduced in the next subsections. More details about the negotiation support part are given in Rosa Gil s PhD thesis [49] and related papers [48, 26].

[16]http://dmag.upf.edu/newmars

First phase Once the customer has chosen his representative agent, it is created and all the necessary data is loaded in the expert system. The metadata that models the negotiated offer and its context is loaded together with all the ontologies that define the concepts used by the metadata.

As has been already shown, all is expressed in RDF and OWL. In order to operate with JESS, all the metadata and ontologies are imported using OWLJessKB [73]. After that, the negotiation protocol and policies are also loaded. They are modeled as a set of rules in JESS format.

The protocol rules govern the timing of the different negotiation phases. On the other hand, the policy rules support the decision process of the agent. For instance, buy or sell only when a condition about price or duration is achieved.

This is an important feature because it allows us to determine important contract parameters as duration, prices and so on. Thus, we get a dynamic negotiation mechanism because negotiation policies can be easily changed and configured.

Second phase In this phase, the negotiation is finally carried out. The customer agent contacts the agent that is in charge of the offer negotiation. This is done using the information captured in the initial offer. The "agent" case role of the "Offer" action identifies the corresponding agent using a JADE identifier.

The content provider agent that is responsible for negotiating the offer is the representative of the content provider that made the offer. It is ready to handle negotiations and pre-configured with the desired negotiation policy.

When it is contacted, it retrieves the negotiated offer from the NewMARS broker. It is loaded together with the received counteroffer and the required ontologies in the JESS engine that governs its behaviour.

The customer will then use the customer agent as the intermediary between him and the content provider agent. The customer agent can be more or less interactive, i.e. more or less autonomous. On the other hand, the content provider agent is autonomous and thus it takes decisions completely on its own, as specified in its negotiation policy.

The negotiation process goes on through the corresponding protocol as a series of offers and counteroffers. The outcome can be an agreement if both parts agree on the offer conditions. These conditions will then constitute the license that is digitally signed by both parts. On the other hand, the negotiation process can fail if any part leaves the process.

11.4.4 Conclusions

The main conclusion from the NewMARS development has been the great benefits that can be obtained from a knowledge-oriented application. A high module independence based on the particular semantics has been achieved. This allows employing the same techniques for different domains by only adapting the conceptual framework, i.e. the ontologies that define the metadata structure.

For instance, in order to check NewMARS semantic capabilities, it has been also used with third party metadata. Concretely, it has been fed with RDF metadata from the MusicBrainz website[17]. This project has its own ontology for the music domain, i.e. album, track, artist, etc. The only effort necessary in order to make NewMARS manage resources annotated with MusicBrainz metadata has been to connect its ontology with the NewMARS ontological framework.

This has been easy thanks to the foundation of NewMARS ontological framework in Copyright Ontology, a quite generic conceptualisation. Therefore, NewMARS can be easily configured to manage rights for any kind of copyrighted content.

[17] http://www.musicbrainz.org/

Part IV

Conclusion

Chapter 12

Conclusions

From the conceptualisation, implementation and evaluation work, it is possible to conclude that the main hypotheses have been correctly posed. The Semantic Web tools allow facing the Digital Rights Management problem with an additional set of features, in comparison to the more common XML tools. These new features make possible to develop more complete domain models and facilitate the implementation of DRM applications.

The limitations of the current approaches are overcome by the ontology-based approach. The increased expressivity level allows formalising a greater part of the domain semantics, in this case of the copyright domain. Once formalised, this semantics are easier to implement, as it has been shown in the Evaluation chapter 11, e.g. the benefits of semantic queries versus syntactic ones like XPath.

Another consequence of grammar-based RELs expressiveness limitations is that they do not take into account the legal framework where the DRM systems that employ them operate. This is because the copyright legal framework is too complicated to be captured by syntax-based tools. However, as it has been shown in the Conceptualisation chapter 9, it is possible to develop a conceptual framework for both the legal and DRM parts of the domain, i.e. the contributed Copyright Ontology.

Moreover, the model expressivity solves the current RELs problem with the level of detail they have to support in order to cope with usage control. They have to specify every possible allowable usage in their grammar, or at least provide expansion points for this. Ontologies provide generalisation mechanisms and they can cope simultaneously with many levels of detail, as it is required by the rights situation at hand.

This kind of complex conceptual models can also be implemented using syntactic-approaches, but most of their semantics is lost from the formalisation point of view. The consequence of these limitations uses to be that the legal part of the model is not included in the implemented version and even ignored during the corresponding conceptualisation phase, as it is has been shown in the Rights Expression Languages chapter 5.2.

On the contrary, if the conceptual model is implemented using ontologies, a greater part of the model semantics can be captured. A Semantic Web based implementation of this model has been carried out. It is based on the Web Ontology Language (OWL) and, although it is not the more expressive ontology language available, it has shown its practical benefits.

The OWL implementation of the Copyright Ontology integrates the legal and DRM aspects. Moreover, its particular approach makes an extensive use of OWL classes as patterns for rights, licenses and usages modelling. With this approach, it is possible to reduce checking uses against rights and licenses to computing class subsumption and instance classification. These computations are tractable and decidable in the context of Description Logics, a family of logics that includes OWL-DL, the OWL sublanguage employed during the implementation.

Therefore, the implementation can be based on existing Description Logic reasoners, which facilitate the development of semantics-based DRM systems. This implementation approach has been sketched in the Implementation chapter 10 and it has been put into practice in the Semantic DRM System section 11.4. Other benefits of the Semantic Web approach to DRM are shown in the Evaluation chapter. This chapter also includes the ontology versions of the main REL initiatives that allow performing an extensive evaluation of the Copyright Ontology in the context of real-world DRM requirements.

These practical results have been contributed to the MPEG-21 standardisation initiative for a DRM framework, as it is detailed in the Publications section 12.1. Moreover, in this section it is shown how the entire contribution has been validated in the forum of relevant international conferences and standardisation bodies.

To conclude, and from a more general point of view, it has been possible to observe the potential of Knowledge Representation in the Web context. The Semantic Web approach is being applied to very diverse application domains, e.g. e-science, e-commerce, business integration, recommendation systems or social networks. This work shows that it is also useful in the Digital Rights Management domain.

12.1 Publications

Just from the beginning, the research work has been validated against the research community in the context of relevant conferences and standardisation initiatives.

12.1.1 Standardisation Contributions

The contributions to standardisation bodies are centred on the MPEG-21[1] standardisation process. This initiative tries to define a complete content management framework that includes a complete digital rights management solution. The first contributions to MPEG-21 related to this work is a previous version of the Copyright Ontology, IPROnto[2], contributed as a complete solution for the MPEG-21 call for contributions for a Rights Expression Language and Rights Data Dictionary [82]:

- **IPROnto - Intellectual Property Rights Ontology**
  DMAG Contribution to MPEG-21, Rights Data Dictionary and Rights Expression Language call for proposals, 2001

After this initial contribution, there have been many others related to the use of Semantic Web tools for implementing MPEG-21 reference software. Moreover, it has been possible to evaluate the standard using a Web ontology, RDDOnto[3]. The validation of the RDD ontology has shown many inconsistencies in the standard that have caused a standard corrigendum [5] and have recently opened another revision of the standard.

- **RDDOnto, Rights Data Dictionary Ontology**
  García, R.; Delgado, J.; Rodríguez, E. and Llorente, S.
  ISO/IECJTC1/SC29/WG11/M10124, October 2003, Brisbane, Australia
- **RDDOnto, Rights Data Dictionary Ontology Version 2**
  García, R.; Delgado, J.; Rodríguez, E.; Llorente, S. and Gallego I.
  ISO/IECJTC1/SC29/WG11/M10423, December 2003, Hawaii, United States
- **MPEG REL and RDD Ontologies API**
  García, R.; Delgado, J. and Gallego, I.:
  ISO/IECJTC1/SC29/WG11/M10702. March 2004, Munich, Germany

[1] http://www.chiariglione.org/mpeg/standards.htm
[2] http://dmag.upf.edu/ontologies/ipronto
[3] http://dmag.upf.edu/ontologies/mpeg21ontos

- **RELOntos (REL Ontologies)**
  García, R.; Delgado, J. and Gallego, I.
  ISO/IECJTC1/SC29/WG11/M10703. March 2004, Munich, Germany

- **An ontology approach for REL use of rights data dictionaries**
  Gil, R.; García, R.; Delgado, J. and Rodríguez, E.
  ISO/IECJTC1/SC29/WG11/M12060. April 2005, Busan, Korea

- **MPEG 21 RDD spec vs MPEG 21 Requirements and Vision documents Review**
  Gauvin, M.; Delgado, J.; García, R. and Rodríguez, E.
  ISO/IECJTC1/SC29/WG11/M11875. April 2005, Busan, Korea

- **Ontological Analysis of the MPEG-21 Rights Data Dictionary (RDD)**
  García, R.; Delgado, J. and Rodríguez, E.
  ISO/IECJTC1/SC29/WG11/M12495. October 2005, Nice, France

- **Preliminary Results of RDD CE for Expressing "Work"**
  Gauvin, M.; García, R.; Delgado, J. and Rodríguez, E.
  ISO/IECJTC1/SC29/WG11/M12511. October 2005, Nice, France

12.1.2 Refereed publications

These are the publications resulting from the contribution of results related to this work to international conferences and workshops. They are related to different aspects of this work: the Copyright Ontology, the Semantic Web approach to DRM in general, semantic DRM systems, agents-mediated copyright negotiation and the XML Semantics Reuse methodology.

Copyright Ontology

These publications are mainly related to the Copyright Ontology. The Copyright Ontology has been discussed in the main international conferences and workshops dealing with the legal ontologies, legal knowledge, regulatory ontologies and Semantic Web topics.

- **A web ontologies framework for digital rights management**
  Gil, R.; García, R. and Delgado, J.
  Journal of Artificial Intelligence and Law. Special issue on Legal Ontologies and Artificial Intelligence Techniques
  Under review, 2005

- **An interoperable framework for IPR using web ontologies**
  Gil, R.; García, R. and Delgado, J.
  Legal Ontologies and Artificial Intelligence Techniques, LOAIT 2005
  IAAIL Workshop Series, pp. 135-148. Wolf Legal Publishers, 2005
  ISBN 90-5850-504-9

- **IPROnto: An Ontology for Digital Rights Management**
  Delgado, J.; Gallego, I.; Llorente, S. and García, R.
  16th Annual Conference on Legal Knowledge and Information Systems, JURIX 2003
  Frontiers in Artificial Intelligence and Applications, Vol. 106, IOS Press, 2003
  ISBN 1-58603-398-0

- **Regulatory Ontologies: An Intellectual Property Rights approach**
  Delgado, J.; Gallego, I.; Llorente, S. and García, R.
  Workshop on Regulatory Ontologies and the Modeling of Complaint Regulations, WORM CoRe 2003
  Lecture Notes in Computer Science, Vol. 2889, pp 621 - 634. Springer-Verlarg, 2003
  ISBN 3-540-20494-6

- **IPROnto - Intellectual Property Rights Ontology**
  Delgado, J.; Gallego, I.; García, R. and Gil, R.
  Poster in International Semantic Web Conference (ISWC), Sardinia, 2002

Semantic Web Approach to DRM

These publications include contributions mainly related to the general benefits of the Semantic Web approach for existing rights expression languages. They have been contributed to international conferences and workshops dealing with the legal ontologies and e-commerce fields. There is also a contribution to the ODRL Workshop, which is the main discussion forum for research related to this REL.

- **An Ontological Approach for the Management of Rights Data Dictionaries**
  García, R. and Delgado, J.
  18th Annual Conference on Legal Knowledge and Information Systems, JURIX 2005
  To be published

- **Formalising ODRL Semantics using Web Ontologies**
  García, R.; Gil, R.; Gallego, I. and Delgado, J.
  Open Digital Rights Language Workshop, ODRL'2005

- **Use of Semantic Tools for a Digital Rights Dictionary**
  Delgado, J.; Gallego, I. and García, R.
  5th International Conference on Electronic Commerce and Web Technologies, EC-WEB 2004
  Lecture Notes in Computer Science, Vol. 3182, pp 338 - 347. Springer-Verlarg, 2004
  ISBN 3-540-22917-5

Semantic DRM Systems

The following publications are mainly related to the development of semantics-enabled DRM systems. They have been presented in the context of international conferences and workshops dealing with sematic information systems, e-commerce and Semantic Seb.

- **Architecture of a Semantic XPath Processor. Application to Digital Rights Management**
  Tous, R.; García, R.; Rodríguez, E. and Delgado, J.
  E-Commerce and Web Technologies: 6th International Conference, EC-Web 2005
  Lecture Notes in Computer Science, Vol. 3590, pp 1 - 10. Springer-Verlarg, 2005
  ISBN 3-540-28467-2
- **Intellectual Property Rights Management using a Semantic Web Information System**
  García, R., Gil, R. and Delgado, J.
  OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2004
  Lecture Notes in Computer Science, Vol. 3291, pp 689 - 704. Springer-Verlarg, 2004
  ISBN 3-540-23662-7
- **Brokerage of Intellectual Property Rights in the Semantic Web**
  García, R. and Delgado, J.
  Proceedings Semantic Web Working Symposium, pp 245-260, SWWS, Stanford, 2001

Agents-mediated Copyright Negotiation

The main research area of these publications is the area of artificial agents. They are applied in order to enable automatic and assisted negotiation of copyrighted works in the context of semantic DRM systems. They have been contributed to different editions of one of the main international conferences on agents for e-commerce applications.

- **Delivery context negotiated by mobile agents using CC/PP**
  Gil, R., García, R. and Delgado, J.

Mobile Agents for Telecommunication Applications, MATA 2003
Lecture Notes in Computer Science, Vol. 2881, pp 99 - 110. Springer-Verlarg, 2003
ISBN 3-540-20298-6

- **An Architecture for Negotiation with Mobile Agents**
  Delgado, J.; Gallego, I.; García, R. and Gil, R.
  Mobile Agents for Telecommunication Applications, MATA 2002
  Lecture Notes in Computer Science, Vol. 2521, pp 21-31. Springer-Verlarg, 2002
  ISBN 3-540-00021-6

Chapter 13

Future Work

The future plans for this work can be classified in three lines. First, to complete some of the research and development lines already initiated. This future work line is mainly related to the mapping of differente REL initiatives to the Copyright Ontology. Second, to develop a semantics-enabled DRM system that fully exploits the Semantic Web approach to DRM. And third, to add new features to the semantics-enabled DRM system build up by the conjuntion of all the contributions of this work.

13.1 Mappings to the Copyright Ontology

As it has been said in the Evaluation chapter 11, it is still necessary to complete the mappings from the MPEG-21 REL and RDD Ontologies and from the ODRL Ontology to the Copyright Ontology. This mappings should be complemented with a mapping from Creative Commons (CC). Although CC is a very simple rights expression language based on a simple RDF Schema, it is the more widely used one nowadays. The main Creative Commons features that seem the reason for its success are its simplicity, its inclusion of copyright terms and the availability of human-readable versions of the licenses in natural language.

13.2 Semantic DRM System

The NewMARS[1] web application is the current implementation of the Copyright Ontology framework in order to build a semantics-enabled Digital Rights Management System. As it has been shown in the Semantic DRM System section 11.4, it is a quite complete implementation of a Semantic DRMS. However, there are some functionalities that are pending a proper implementation. The main caveat is to build it on top of a semantic metadata store that scales up to great amounts of metadata and which provides a complete solution for OWL-DL reasoning and Semantic Web Rules support.

13.3 New Features

A part from the previous future lines that are geared toward completing already initiated work features, some new features, which would be desirable for a Semantic DRM System, have been detected. First of all is to improve the security and trust of the Semantic DRMS. Another interesting point is to mimic one of the Creative Commons features, i.e. to provide a natural language version of the rights expressions.

13.3.1 Security and Trust

The inclusion of security issues that transform the e-commerce application into a trusted one has not been initiated yet. This requires the application of Web of trust ideas and Public Key Infrastructure technologies. The primary intention is requiring that each actor taking part in the systems have its own digital certificate. This certificate, with its corresponding private key, will be used for digitally signing all the statements done by this actor (agreements, offers, assertions. . .) so responsibility can be tracked later and even produce contracts.

The plan is to apply digital signatures at the level of the RDF graph segmentation already employed in NewMARS in order to facilitate browsing metadata, as it is detailed in the corresponding Metadata Retrieval section 11.4.3. A similar approach has been already applied in the DBin project [111].

[1] http://dmag.upf.edu/newmars

13.3.2 Controlled Natural Language Interface

Another interesting feature is to provide a natural language version of the rights expressions in order to facilitate human-user interaction with the Semantic DRM System. This is valuable feature of the Creative Commons initiative. However, in the CC case, the REL is quite simple and the human-readable licenses are not generated autommatically, they are generated a priori and they just provide the legal framework for the license, i.e. they are not personalised for the current copyrighted content.

The idea is to take profit from the linguistic foundations of the Copyright Ontology, i.e. the use of verbs and case roles as the modelling building blocks. Due to this linguistic base, it is possible to generate Controlled Natural Languages [Schwitter05] from the Copyright Ontology expressions. Many different natural languages can be supported. Starting from a mapping from the Copyright Ontology to the lexical resource Wordnet [30], it is possible to autommatically retrieve the lexemes corresponding to Copyright Ontology concepts in English but also in other languages, for instance those present in EuroWordnet [114].

Bibliography

[1] N. Juristo A. Gómez-Pérez and J. Pazos. Evaluation and assessment of knowledge sharing technology. In N. J. Mars, editor, *Towards Very Large Knowledge Bases*, pages 289–296. IOS Press, 1995. 114

[2] M. Alavi and D. E. Leidner. Knowledge management and knowledge management systems: Conceptual foundations and research issues. *Management Information Systems Quarterly*, 25(1):107–136, 2001. 31

[3] B. Amann, C. Beer, I. Fundulak, and M. Scholl. Ontology-based integration of xml web resources. In *Proceedings of the 1st International Semantic Web Conference (ISWC 2002)*, pages 117–131. Springer-Verlag, 2002. 116

[4] American National Standard, ISO/JTC1/SC32/WG2. *Conceptual Graph Standard*, 2001. Proposed draft. 29

[5] C. Barlas and G. Rust. Rights data dictionary, technical corrigendum 1. Technical report, ISO/IEC JTC1/SC29/WG11, 2005. 222, 246

[6] Chris Barlas. The imprimatur project. In Klaus Brunnstein and Peter Paul Sint, editors, *Intellectual Property Rights and New Technologies, Proceedings of the KnowRight '95 Conference*, volume 82, pages 264–272, Wien, Austria, 1995. Austrian Computer Society. 228

[7] David Becket. Rdf/xml syntax specification. W3c recommendation, RDF Core Working Group, World Wide Web Consortium, 2004. 55, 233

[8] F. Bellifemine, G. Caire, A. Pogg, and G. Rimassa. Jade - a white paper. Technical Report 3, Telecom Italia Lab, EXP Online, 2003. 233

[9] Rodrigo Bercovitz et al. *The Copyright Handbook*. Tirant Lo Blanch, 2003. (in Spanish). 123

[10] Tim Berners-Lee. *Weaving the Web*. HarperBusiness, New York, 2000. 42

[11] Tim Berners-Lee. Cwm: A general purpose data processor for the semantic web. Technical report, Word Wide Web Consortium, 2005. 60

[12] Tim Berners-Lee. Notation3: Logic and rules on rdf. Technical report, Word Wide Web Consortium, 2005. 59

[13] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, May 2001. 2, 42, 49

[14] G. Booch, J. Rumbauch, and I. Jacobson. *The Unified Modeling Language user guide*. Addison Wesley Longman Inc., 1999. 108

[15] R. J. Brachman. On the epistemological status of semantic networks. In N. V. Findlet, editor, *Associative Networks: Representation and Use of Knowledge by Computers*, pages 3–50. Academic Press, 1979. 11

[16] J. Breuker. Constructing a legal core ontology: Lri-core. In *Workshop on Ontologies and their Applications*, Sao Luis, Maranhao, Brazil, 2004. 25, 185

[17] D. Brickley and R. V. Guha. Rdf vocabulary description language 1.0: Rdf schema. W3c recommendation, RDF Core Working Group, World Wide Web Consortium, 2004. 56

[18] G. Chierchia and S. McConnell-Ginet. *Meaning and Grammar. An introduction to semantics*. MIT Press, second edition, 2000. 16

[19] O. Corcho, M. Fernández, A. Gómez-Pérez, and A. López-Cima. Building legal ontologies with methontology and webode. In R. Benjamins, P. Casanovas, J. Breuker, and A. Gangemi, editors, *Law and the Semantic Web*, number 3369 in LNAI, pages 142–157. Springer-Verlag, 2005. xviii, 109, 111

[20] Karen Coyle. Rights expressions languages. Technical report, Library of Congress, February 2004. 69

[21] I. Cruz, H. Xiao, and F. Hsu. An ontology-based framework for xml semantic integration. In *Eighth International Database Engineering and Applications Symposium*, Coimbra, Portugal, 2004. 116

[22] R. Van de Walle and Ian Burnett. *The MPEG-21 Book*. John Wiley & Sons, Chichester, UK, 2005. 1, 68, 87

[23] M. Dean and G. Schreiber. Owl web ontology language reference. W3c recommendation, Web Ontology Working Group, World Wide Web Consortium, 2004. 58, 188, 221, 233

[24] M. Dekkers and S. Weibel. State of the dublin core metadata initiative. *D-Lib Magazine*, 9(4), April 2003. 73, 218, 231

[25] G. Deleuze and F. Guattari. *A thousand plateaus: capitalism and schizophrenia*. University of Minnesota Press, 1987. Translation B. Massumi. 53

[26] J. Delgado, I. Gallego, R. García, and R. Gil. An architecture for negotiation with mobile agents. In A. Karmouch, T. Magedanz, and J. Delgado, editors, *Mobile Agents for Telecommunication Applications: 4th International Workshop*, volume 2521 of *LNCS*, pages 21–31. Springer-Verlag, 2002. 197, 240

[27] J. Delgado, I. Gallego, R. García, R. Gil, E. Peig, and R. Tous. Intellectual property rights ontology (ipronto). Contribution to MPEG-21 RDD-REL CfP M7675, Moving Picture Experts Group, 2001. 2

[28] J. Delgado, I. Gallego, and J. Polo. Electronic commerce of multimedia services. In *Multimedia Modelling*, pages 97–110. World Scientific Publishing, 1999. 228

[29] J. P. Dick. *A conceptual, case-relation representation of text for intelligent retrieval*. PhD thesis, University of Toronto, Canada, 1991. 159

[30] C. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, 1998. 253

[31] D. Fensel and C Bussler. The web service modeling framework wsmf. Technical report, Web Service Modeling Framework Initiative, 2002. White paper. 39

[32] M. Fernández-López. Overview of methodologies for building ontologies. In *Proc. Workshop on Ontologies and Problem-Solving Methods*, 1999. 107

[33] M. Fernández-López, A. Gómez-Pérez, and N. Juristo. Methontology: from ontological art towards ontological engineering. In *Proc. Symposium on Ontological Engineering of AAAI*, 1997. 108

[34] M. Fernández-López, A. Gómez-Pérez, A. Pazos-Sierra, and J. Pazos-Sierra. Building a chemical ontology using methontology and the ontology design environment. *IEEE Intelligent Systems and their applications*, pages 37–46, Jan/Feb 1999. 112, 114

[35] M. Fernández-López, A. Gómez-Pérez, and M. D. Rojas. Ontologies' crossed life cycles. In *Proc. International Conference in Knowledge Engineering and Management*, volume 1937 of *LNAI*, pages 65–79. Springer-Verlag, 2000. 110

[36] R. Fielding et al. *Hypertext Transfer Protocol - HTTP/1.1*. Network Working Group, The Internet Society, 1999. RFC 2616. 34

[37] FIPA. Fipa acl message structure specification. FIPA Agent Communication Language Specifications XC00061, Foundation for Intelligent Physical Agents, 2002. 233, 235

[38] P. Flichy. *Dynamics of Modern Communication: The Shaping and Impact of New Communications Technologies*. Sage Publications, 1995. Translation L. Libbrecht. 43

[39] E. Friedman-Hill. *Jess in Action: Rule-Based Systems in Java*. Manning Publications Co., 2003. 233, 240

[40] I. Gallego, J. Delgado, and J. Acebron. Distributed models for brokerage on electronic commerce. In *Trends in distributed systems for electronic commerce*, pages 129–140. Springer-Verlag, 1998. 228

[41] A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, and L. Schneider. Sweetening ontologies with dolce. In A. Gómez-Pérez and V.R. Benjamins, editors, *Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, 13th International Conference, EKAW 2002*, pages 166–181. Springer-Verlag, 2002. 25, 185, 220

[42] R. García and O. Celma. Semantic integration and retrieval of multimedia metadata. In *Proceedings of the ISWC 2005 Workshop on Knowledge Markup and Semantic Annotation (Semannot'2005)*. CEUR Workshop Proceedings, 2005. 115, 195, 231

[43] R. García and J. Delgado. Brokerage of intellectual property rights in the semantic web. In I. F. Cruz, S. Decker, J. Euzenat, and D. L. McGuinness, editors, *Proc. Semantic Web Working Symposium*, pages 245–260, Stanford University, California, 2001. 226, 228

[44] R. García, J. Delgado, and I. Gallego. Rddonto, rights data dictionary ontology version 2. Technical Report M10423, ISO/IEC JTC1/SC29/WG11, Hawaii, United States, 2003. 221

[45] R. García, J. Delgado, and E. Rodríguez. Ontological analysis of the mpeg-21 rights data dictionary (rdd). Technical Report M12495, ISO/IEC JTC1/SC29/WG11, Nice, France, 2005. 223

[46] R. García, R. Gil, and J. Delgado. Intellectual property rights management using a semantic web information system. In R. Meersman and Z. Tari, editors, *OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2004*, volume 3291 of *LNCS*, pages 689–704. Springer-Verlag, 2004. 226

[47] M. Genesereth. Knowledge interchange format. Proposed Draft NCITS.T2/98-004, American National Standard, 1998. 29

[48] R. Gil, R. García, and J. Delgado. Delivery context negotiated by mobile agents using cc/pp. In E. Horlait, T. Magedanz, and R. H. Glitho, editors, *Mobile Agents for Telecom Applications, (MATA'03)*, volume 2881 of *LNCS*, pages 99–110. Springer-Verlag, 2003. 197, 240

[49] Rosa M. Gil. *Agents negotiating in a semantic web architecture*. PhD thesis, Technology Department, Universitat Pompeu Fabra, 2005. 240

[50] A. Gómez-Pérez. Knowledge sharing and reuse. In J. Liebowitz, editor, *Handbook of Expert Systems*. CRC, 1998. 112

[51] A. Gómez-Pérez, M. Fernández, and O. Corcho. *Ontological Engineering*. Springer-Verlag, London, 2004. 107

[52] A. Gómez-Pérez and M. D. Rojas. Ontological reengineering and reuse. In *Proc. European Knowledge Acquisition Workshop*, 1999. 114

[53] Kurt Godel. *On Formally Undecidable Propositions of Principia Mathematica and Related Systems*. Dover Publications, 1992. Translation B. Meltzer. 15, 20

[54] Nicola Guarino and Christopher Welty. Evaluating ontological decisions with ontoclean. *Communications of the ACM*, 45(2):61–65, 2002. 115

[55] S. Tabet H. Boley and G. Wagner. Design rationale of ruleml: A markup language for semantic web rules. In I. F. Cruz, S. Decker, J. Euzenat, and D. L. McGuinness, editors, *Proc. Semantic Web Working Symposium*, pages 381–402, Stanford University, California, 2001. 59

[56] J. Hendler. Agents and the semantic web. *IEEE Intelligent Systems*, 16(2), March-April 2001. 2, 49

[57] F. Heylighen. World-wide web: a distributed hypermedia paradigm for global networking. In *SHARE Europe: The Changing Role of IT in Business*, pages 355–368, 1994. 33

[58] F. Heylighen. Towards a global brain. integrating individuals into the world-wide electronic network. In *Der Sinn der Sinne*. Steidl Verlag, 1997. 45, 50

[59] F. Heylighen and J. Bollen. The world-wide web as asuper-brain: from metaphor to model. In *Cybernetics and Systems '96*, pages 917–922. Austrian Society for Cybernetics, 1996. 43, 50

[60] I. Horrocks and P. F. Patel-Schneider. Optimising propositional modal satisfiability for description logic subsumption. volume 1476 of *LNAI*. Springer-Verlag, 1998. 29

[61] Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, and Mike Dean. Swrl: A semantic web rule language combining owl and ruleml. W3c member submission, World Wide Web Consortium, 2004. 59

[62] Marc-Philippe Huget. Agent uml notation for multiagent system design. *IEEE Internet Computing*, 8(4):63–71, July/August 2004. 108

[63] J. Hunter. Enhancing the semantic interoperability of multimedia through a core ontology. *IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on Conceptual and Dynamical Aspects of Multimedia Content Description*, 13(1):49–58, 2003. 2

[64] R. Iannella. Open digital rights language (odrl), version 1.1. W3c note, World Wide Web Consortium, 2002. 1, 66, 202, 206

[65] ISO. Multimedia framework (mpeg-21) – part 1: Vision, technologies and strategy. Technical report, ISO/IEC TR 21000-1:2004, 2004. 150

[66] ISO. Multimedia framework (mpeg-21) – part 5: Rights expression language. Technical report, ISO/IEC TR 21000-5:2004, 2004. 215

[67] ISO. Multimedia framework (mpeg-21) – part 6: Rights data dictionary. Technical report, ISO/IEC TR 21000-6:2004, 2004. 151, 217, 221, 222

[68] R. Jasper and M. Uschold. Enabling task-centered knowledge support through semantic metadata. In D. Fensel, J. Hendler, H. Lieberman, and W. Wahlster, editors, *Semantic Web Technology*. MIT Press, 2001. 46

[69] Frank van Harmelen Jeen Broekstra, Arjohn Kampman. Sesame: An architecture for storing and querying rdf and rdf schema. In *Proceedings of the First International Semantic Web Conference, ISWC 2002*, volume 2342 of *LNCS*, pages 54–68. Springer-Verlag, 2002. 238

[70] G. Karvounarakis, A. Magkanaraki, S. Alexaki, V. Christophides, D. Plexousakis, M. Scholl, and K. Tolle. Rql: A functional query language for rdf. In P. M. D. Gray, L. Kerschberg, P. J. H. King, and A. Poulovassilis, editors, *The Functional Approach to Data Management: Modelling, Analyzing and Integrating Heterogeneous Data*, LNCS, pages 435–465. Springer-Verlag, 2004. 238

[71] M.C.A. Klein. Interpreting xml documents via an rdf schema ontology. In *Proceedings of the 13th International Workshop on Database and Expert Systems Applications (DEXA 2002)*, pages 889–894, 2002. 116, 118

[72] G. Klir. *Facets of Systems Science*. Plenum Pub Corp, 1992. 5

[73] J. Kopena and W. Regli. Damljesskb: A tool for reasoning with the semantic web. *IEEE Intelligent Systems*, 18(3):74–77, 2003. 241

[74] R. A. Kowalski. *Logic for Problem Solving*. North Holland Publishing Co., 1986. 15

[75] P. Kruchten. *The Rational Unified Process: An Introduction*. Addison Wesley, 1998. 108

[76] L. Lakshmanan and F. Sadri. Interoperability on xml data. In *Proceedings of the 2nd International Semantic Web Conference (ICSW 03)*. Springer-Verlag, 2003. 116

[77] D. A. Leishman. Principled analogical tool: Based on evaluations of partial correspondences over conceptual graphs. Master's thesis, University of Calgary, Canada, 1989. 31

[78] Lawrence Lessig. *The Creative Commons*. RBL, Tokyo, 2003. 2, 66

[79] C. J. Lofting. The neuro-cognitive and emotional roots of mathematics. Web page. 9

[80] G. Lausen M. Kifer and J. Wu. Logical foundations of object-oriented and frame-based languages. *Journal of the Association for Computing Machinery*, 1995. 15

[81] David Martin et al. Bringing semantics to web services: The owl-s approach. In Jorge Cardoso and Amit P. Sheth, editors, *Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004)*, volume 3387 of *LNCS*. Springer, 2004. 61

[82] MPEG. Call for proposals for a rights data dictionary and a rights expression language. Technical Report W4335, ISO/IEC JTC1/SC29/WG11, Sydney, Australia, 2001. 2, 246

[83] Network Working Group, The Internet Society. *Uniform Resource Identifiers (URI): Generic Syntax*, 1998. RFC 2396. 34

[84] Ian Niles and Adam Pease. Towards a standard upper ontology. In C. Welty and B. Smith, editors, *FOIS '01: Proceedings of the international conference on Formal Ontology in Information Systems*, pages 2–9, New York, NY, USA, 2001. ACM Press. 24, 185

[85] I. Nonaka and H. Takeuchi. *The knowledge-creating company*. Oxford University Press, 1995. 6

[86] J. Novak and D. Gowin. *Learning How to Learn*. Cambridge University Press, 84. 9

[87] Jeff Z. Pan. *Description Logics: Reasoning Support for the Semantic Web*. PhD thesis, School of Computer Science, The University of Manchester, Oxford Rd, Manchester M13 9PL, UK, 2004. 189, 211, 222, 224

[88] Jeff Z. Pan and Ian Horrocks. Owl-eu: Adding customised datatypes into owl. In Jérôme Euzenat and Asunción Gómez-Pérez, editors, *Proc. of Second European Semantic Web Conference (ESWC 2005)*, volume 3532 of *LNCS*, pages 153–166. Springer-Verlag, 2005. 197

[89] Peter F. Patel-Schneider. A proposal for a swrl extension to first-order logic. Technical report, DARPA DAML Program, 2004. 60, 193

[90] P.F. Patel-Schneider and J. Simeon. The yin/yang web: Xml syntax and rdf semantics. In *Proceedings of the 11th International World Wide Web Conference (WWW02)*, pages 443–453, 2002. 116

[91] C. S. Peirce. *Existential Graphs, Manuscript 514*. 1909. 29

[92] C. S. Pierce. *The Collected Papers of Charles Sanders Peirce*. Harvard University Press, 1935. Editors C. Hartshorne and P. Weiss. 18

[93] M. Polanyi. The tacit dimension. In L. Prusak, editor, *Knowledge in Organizations*, pages 135–146. Butterworth-Heinemann, 1997. 5

[94] Yuzhong Qu, Xiang Zhang, and Huiying Li. Orel: an ontology-based rights expression language. In *WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 324–325, New York, NY, USA, 2004. ACM Press. 2

[95] H. Schrobe R. Davis and P. Szolovits. What is knowledge representation? *AI Magazine*, 14(1):17–33, 1993. 10

[96] R R. Studer, R. Benjamins, and D. Fensel. Knowledge engineering: principles and methods. *Data and knowledge engineering*, 25:161–197, 1998. xvii, 23

[97] D. Reynolds. Amateur introduction to description logics. Technical report, Hewlett-Packard Laboratory, 2001. xvii, 30

[98] M. D. Rojas. Ontologias de iones monoatomicos en variables fisicos del medio ambiente. Master's thesis, Facultad Informatica, Universidad Politecnica de Madrid, Spain, 1998. 112, 115

[99] Stuart Russell and Peter Norvig. *Artificial Intelligence, a modern approach*. Prentice Hall, 1995. 19

[100] Godfrey Rust and Mark Bide. The <indecs> metadata framework: Principles, model and data dictionary. Technical Report WP1a-006-2.0, Indecs Framework Ltd, June 2000. © Indecs Framework Ltd. 150

[101] P. Salembier and J. Smith. Overview of mpeg-7 multimedia description schemes and schema tools. In B. S. Manjunath, P. Salembier, and T. Sikora, editors, *Introduction to MPEG-7: Multimedia Content Description Interface*. John Wiley & Sons, Chichester, 2002. 79, 231

[102] David J. Schultz et al. Ieee standard for developing software life cycle processes. IEEE Std 1074-1997, The Institute of Electrical and Electronics Engineers, New York, USA, 1997. 108

[103] J. E. Simpson. *XPath and XPointer. Locating Content in XML Documents*. O'Reilly, 2002. 116

[104] R. Smith. What's required in knowledge technologies - a practical view. In *Knowledge Technologies*, 2001. 46

[105] John F. Sowa. *Principles of Semantic Networks: Explorations in the Representation of Knowledge*. Representation and Reasoning. Morgan Kaufmann, 1991. 218

[106] John F. Sowa. *Knowledge Representation. Logical, philosophical and computational foundations*. Brooks Cole Publishing Co., 2000. 10, 22, 159

[107] Mark Stefik. Shifting the possible: How trusted systems and digital property rights challenge us to rethink digital publishing. *Berkeley Technology Law*, 12(1), 1997. 68

[108] D. Stenmark. Information vs. knowledge: The role of intranets in knowledge management. In *Proceedings of HICSS*. IEEE Press, 2002. 8

[109] R. Tous, R García, E. Rodríguez, and J. Delgado. Arquitecture of a semantic xpath processor. application to digital rights management (drm). In K. Bauknecht, B. Pröll, and H. Werthner, editors, *E-Commerce and Web Technologies: 6th International Conference, EC-Web 2005*, volume 3590 of *LNCS*, pages 1–10. Springer-Verlag, 2005. 118

[110] H. Tsoukas. The firm as a distributed knowledge system: A constructionist approach. *Strategic Management Journal*, 17:11–25, 1996. 5

[111] G. Tummarello, C. Morbidon, P. Puliti, and F. Piazza. The dbin semantic web platform: An overview. In *WWW2005 Workshop on The Semantic Computing Initiative (SeC?05)*, Chiba, Japan, 2005. 252

[112] UDDI Specification Technical Committee, OASIS Consortium. *UDDI Version 3.0*, 2002. 39

[113] M. Uschold and M. Gruninger. Ontologies: principles, methods and applications. *Knowledge Engineering Review*, 11(2), 1996. 112, 114

[114] P. Vossen. *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*. Kluwer Academic Publishers, 1998. 253

[115] W3C HTML Working Group. *XHTML 1.0 The Extensible HyperText Markup Language*, second edition, 2002. W3C Recommendation. 35

[116] W3C Web Services Description Working Group. *Web Services Description Language (WSDL) version 1.2*, 2002. Working Draft. 39

[117] W3C XML Protocol Working Group. *SOAP Version 1.2 Part 0: Primer*, 2002. Working Draft. 40, 233

[118] W3C XML Schema Working Group. *XML Schema Part 0: Primer*, 2001. W3C Recommendation. 38

[119] W3C XSL Working Group. *Extensible Stylesheet Language (XSL) Version 1.0*, 2001. 38, 239

[120] X. Wang, T. DeMartini, B. Wragg, M. Paramasivam, and C. Barlas. The mpeg-21 rights expression language and rights data dictionary. *IEEE Transactions on Multimedia*, 7(3):408–417, 2005. 1, 66, 88

[121] Xin Wang, Guillermo Lao, Thomas DeMartini, Hari Reddy, Mai Nguyen, and Edgar Valenzuela. Xrml - extensible rights markup language. In *XMLSEC '02: Proceedings of the 2002 ACM workshop on XML security*, pages 71–79, New York, NY, USA, 2002. ACM Press. 68

[122] WIPO. *WIPO Intellectual Property Handbook: Policy, Law and Use*. Number 489. WIPO Publications, 2004. 122

[123] F. Yergeau, T. Bray, J. Paoli, C. M. Sperberg-McQueen, and E. Maler. Extensible markup language (xml) 1.0. W3c recommendation, XML Core Working Group, World Wide Web Consortium, 2004. 227