

Overview of the MPEG-21 Media Contract Ontology

Editor(s): Name Surname, University, Country

Solicited review(s): Name Surname, University, Country

Open review(s): Name Surname, University, Country

Víctor Rodríguez-Doncel^a, Jaime Delgado^b, Silvia Llorente^b, Eva Rodríguez^b and Laurent Boch^c

^a *Universidad Politécnica de Madrid (UPM)*

^b *Universitat Politècnica de Catalunya (UPC)*

^c *Radiotelevisione Italiana (RAI)*

Abstract. The MPEG-21 Media Contract Ontology (MCO), a part of the standard ISO/IEC 21000, is an ontology to represent contracts dealing with rights on multimedia assets and intellectual property protected content in general. A core model provides the elements to describe the permissions, obligations and prohibitions exchanged in the clauses of a contract. Specific vocabulary is defined in a model extension to represent the most common rights and constraints in the audiovisual context. Design principles, a methodology and a comparative analysis are given, as well as the practical guidelines to use the standard. A thorough description of the contract creation workflow from an original contract is given, including a sample contract text, the RDF version, the detailed mapping of the most relevant clauses and the reconstructed version. A set of MCO-related tools is described, including (i) the reference software to create and edit MCO contracts; (ii) modules to identify, store, search, validate and deliver MCO contracts and (iii) a tool to convert between the akin Contract Expression Language (CEL) contracts and the MCO contracts and (iv) the actual use of MCO in the Rightsdraw family of services.

Keywords: Standards, Knowledge Representation, Contracts, Multimedia

1. Introduction

The international legal framework on the protection of intellectual property¹ distinguishes between *exploitation rights* and *moral rights*. The creator of a work has some unwaivable and inalienable *moral rights* (like the right to be identified as the author) plus some *exploitation rights* (like the right to distribute copies to the public), which are transferable and of limited duration. The exploitation rights can be transferred with exclusivity or not in exchange of an economic compensation, and their exercise can be limited by different constraints and conditions, typically specified in written contracts.

¹See the *Berne Convention for the Protection of Literary and Artistic Works* (1886), whose text is available online <http://www.wipo.int/treaties/en/ip/berne/>

A *media contract* is an agreement between business parties for the deal of exploitation rights on audiovisual works.

Although the contractual relationships play a central role in the commerce of media assets, they are still largely managed with old-fashioned paradigms, in contrast with other areas of media asset management such as ingestion, annotation, cataloguing, storage, retrieval and distribution, which have begun to benefit from a knowledge representation based on ontologies. The representation of contracts as RDF documents supported by ontologies is an advance in order to deal with the ever increasing amount of material and it facilitates achieving better metadata interoperability and systems integration. Unfortunately, many contracts are still created, signed and kept in a paper form or, if in digital form, in a non-machine-readable environment.

This paper describes and analyzes the MPEG-21 Media Contract Ontology (MCO) [1], a standard based solution for representing contracts —and whose editors are authors of this paper.

MPEG-21 MCO is part of the MPEG-21 specification, formally known as ISO/IEC 21000. MPEG-21 is devoted to the definition of a framework for multimedia delivery and consumption across different networks and devices. In particular, MPEG-21 defines a generic content encapsulation called Digital Item, capable of identifying, structuring and storing different types of digital content (*Digital Item Declaration* and *Identification*). This content can be reliably managed and protected across networks (*Intellectual Property Management and Protection*), and it can be adapted to the heterogeneous usage environment with different networks, terminals and users (*Digital Item Adaptation*). MPEG-21 also defines an expression language for declaring licenses (*Rights Expression Language*) along with a dictionary of terms (*Rights Data Dictionary*), and an ontology (*Media Value Chain Ontology* (MVCO) [2]), capable of representing the intellectual property objects along its life cycle as well as the actors performing any intellectual property-related action over the resources. However, no specific formats had been defined for representing contracts.

The MPEG initiative on specifying formats for media contracts started with the definition of an XML-based expression language: the *Contract Expression Language* (CEL), which eventually became Part 20 of MPEG-21 [3] and is well described in [4]. In parallel, a complementary work was carried out towards an ontology-based representation: the Media Contract Ontology, now Part 21 of MPEG-21. Parts of CEL and MCO reflect the results produced in the European project PrestoPRIME [5] and its audiovisual rights ontology [6], defined as extension of MVCO. CEL and MCO are not competing but complementary and aligned languages, simultaneously conceived to satisfy needs in different contexts, and whose interoperation is possible.

This paper describes the Media Contract Ontology, starting by the design principles, requirements and methodology in Section 2. Section 3 describes the main elements of the ontology and Section 4 details some practical aspects, illustrated with examples and guidance for its use. Section 5 describes the MCO standard Reference Software and some basic applications. Future work and conclusions are outlined in

Section 6.

2. Design Principles, and Requirements and Methodology

The choice of the design principles as well as the methodology to develop the ontology have been strongly mediated by the working framework, i.e., one standardization body with well established procedures, standard timelines, control mechanisms and conflict resolution strategies.

The International Organization for Standardization (ISO), together with the International Electrotechnical Commission (IEC), develops and publishes international standards, by means of Technical Committees (TC) or Joint Technical Committees (JTC), further structured in SubCommittees (SC) and Working Groups (WG) of experts of the various specific areas. The Moving Picture Experts Group (MPEG), officially identified as ISO/IEC JTC1 SC29 WG11, is one of such groups and it sets standards for audio and video compression and transmission as well as for metadata. The following features of the MPEG working framework strongly influenced the design principles and the development methodology for the ontology:

- *High consensus*. The positive vote of at least 5 countries is needed for a standard to be approved and debate is promoted so that differences are solved inasmuch as possible —much appropriated for the development of 'ontologies', which are generally understood as the representation of *consensual* knowledge.
- *Formal procedure*. The standardization process is a well-defined procedure with fixed stages. These steps, which start with a call for requirements, matches well other ontology development methodologies and forces every step to be profusely documented.
- *Working groups*. Any expert can participate in the working groups for the standard edition, which are convened and supervised by the systems group. In practice, heterogeneous domain and technology experts work together.
- *Evolution*. Formal procedures for corrigenda and amendments exist, being also possible second editions. This also favors the continuous development of ontologies.
- *Closed specifications*. ISO standards, in gen-

eral, are not free documents. This feature clashes with the general practice of publishing and disseminating ontologies, however, electronic documents such as an OWL file are typically disclosed freely.

- *Self-containment*. ISO standards are planned to last for years, and introducing dependencies from other sources is discouraged. Consequently, self-contained solutions are preferred over documents making references to external, possibly mutable, resources. This feature goes also against the common practice in the semantic web of reusing others' resources.

The well-established procedures in the standardization process force the adoption of good practices in the ontology development. In particular, the MCO was designed to cover a set of requirements formally specified in 2010 [7] and further refined after a core experiment for mapping text contracts into machine readable documents [8].

The requirements in [7] were not directly imposed on the contract format, but on the functionalities that contract-based services had to provide. The list included 13 functionalities, namely: *authenticate contract* (confirm the identity of the signatories), *check with contract* (verify if a usage request is within a contract), *create contract*, *deliver contract* (possibly adapted with MPEG21 DIA), *identify contract* (with MPEG21 DII), *negotiate contract* (MPEG-M orchestration), *post contract* (possibly with MPEG-U), *present contract* (possibly with MPEG-V), *process contract* (editing the contents) and *request, search, revoke and verify contract*. This specification of contract-based services was translated into requirements for the contract language. It is remarkable that the requirements *did not include any provision for contract breaches and remedies*, nor included advanced reasoning tasks beyond the support to the check with operation (if a requested action under certain conditions falls under the clause of a contract or not). Twelve of the thirteen requirements handled with managing the contract (identifying, storing, etc.), and the “check with” one was a simple operation to be implemented by a service and not to be calculated by means of an ontology reasoner: the most important needs were the preservation of contracts in a structured manner, the easy SPARQL querying of some of the contract features and the easy mapping to existing enterprise systems within the MPEG-21 framework.

Five institutions participated in the experiment,

which consisted of the following steps: (i) a set of contracts was selected with the criteria of including different media types, concerning different parties from different business models and of different length and complexity; (ii) the set of contracts was analyzed in their context by legal experts in the area; (iii) a mapping for these contracts to a candidate machine readable format, first XML then RDF, was made by hand and (iv) the mappings were evaluated with determined criteria.

These criteria included (i) the technical soundness of the output documents (syntactic and semantic); (ii) a comparison of the defined entities with the definition of the referred concepts; (iii) the ability to preserve the contract, granting that reconstructed contracts conveyed sufficient information; (iv) ability to satisfactorily structure and present the contract and (v) ability to link the contract to other external entities (parties, assets). The results of these experiments gave rise to both an XML representation (CEL, Part 20 of MPEG-21) and an RDF representation (MCO, Part 21 of MPEG-21) of the contract; the latter being described in this paper.

3. Related work

The MPEG-21 MCO is not an academic work intending to improve the state of the art. On the contrary, as any standard it sets the recommended practice in its domain and innovations are discouraged. This section describes the related work coming from different areas, collecting the perspective of legal experts, computer scientists and logicians alike, which was considered when developing the ontology.

Within the community of legal experts, the fertile EU project ESTRELLA² produced in the last decade several ontologies of reference in the area, like the ontology of fundamental legal concepts [9] or the interchange format for legal knowledge systems LKIF [10]. These works have a long tradition and they are well founded and supported by subsequent reasoning and argumentation tools; however, the implied meaning of their concepts goes beyond the scope and purpose of the MCO contracts, whose compactness was a design goal: MCO was required not to depend on upper ontologies and it is not relat-

² ESTRELLA, European project for Standardized Transparent Representations in order to Extend Legal Accessibility, 6th European Research Framework Programme

ed at all with these legal ontologies.

In some computer applications, Rights Expression Languages (RELs) and policy languages (like XACML) have played a role governing the access to digital resources. These languages not only allow describing conditions verifiable by a machine, but also declaring permissions and restrictions that cannot be computed. Their structure is suitable for representing contracts as well, with the addition of new vocabulary. In fact, the possible use of RELs for representing contracts has been suggested for ODRL (Open Digital Rights Language [11]) and MPEG-21 REL [12]. However, these extensions do not capture well the structure of the contract nor the relation between the clauses in the text and the operative elements in the electronic contract.

Alternative efforts have focused in making a better logical grounding towards advanced reasoning. The representation of contracts can be done in terms of a formal logic, like the Business Contract Language [13], with powerful reasoning capabilities that include the management of contradictions and conflict resolution procedures. Prisacariu and Schneider propose another contract logic [14], combining deontic logic with a propositional logic of actions to handle complex sequences of actions and tackling contract violations. Other systems have been described with less logical abilities but better expressivity, like Cosmos [15], a contract format described in a well detailed UML object model or Yan's ontology in OWL [16]. A Cosmos contract comprised structures to describe the parties, a fixation (signature) and the performance of activities to be done by the parties, including payments and the provision of goods and services with a given quality of service. Similar elements can be found in the OASIS eContracts [17] specification, which describes a general-purpose, XML-based language. Being a lightweight model, support is given in eContracts for structuring the document in clauses, declaring the relationship with other contracts and signing or encrypting the documents or parts thereof; further refinements being left for specific extensions. MCO is the latest in this tradition, with the narrower scope of representing contracts in the particular domain of the media contracts.

4. Media Contract Ontology

This section describes the Media Contract Ontolo-

gy, which is defined by a core model (described in <http://purl.org/NET/mco-core>), and a first extension for the exploitation of intellectual property rights, (described in <http://purl.org/NET/mco-ipre>). Other extensions, proposed by institutions different from those in the original editing working group, are in the path of standardization but so far they have only working draft status.

4.1. Overview

The Media Contract Ontology (MCO) is an OWL ontology formalizing a vocabulary to represent business contracts in the media content industry. MCO contracts are RDF documents using that vocabulary.

The media contract defines what the parties agree to exchange in terms of rights and obligations. A contract is not bound to deal with a single work nor to constrain the parties to a single role between “*licensor*” and “*licensee*”, although this is the most frequent situation. The most complex part in a typical media contract is the definition of conditions related to the transfer of the exploitation rights. This transfer of rights may be done in exclusivity or not, and may refer to the totality of the exploitation rights or to a part thereof—the latter case being the most common, as it fosters the profitability of the market. Sublicensing is very often explicitly permitted or prohibited, allowing thus the licensee to subsequently grant licenses or not to third parties.

A complex statement defining some exchanged rights, with conditions, on some content or service is also named hereafter an “*operative clause*”, as it defines in which operative context a given action is allowed, prohibited or obligated. The set of operative clauses is called the “*operative part*” of the contract. The fulcrum of an operative clause is the “*deontic expression*”, which encompasses the concepts of *permission*, *prohibition* and *obligation*. As the parties freely agree on the terms of the contract, they actually exchange the promise to respect the rules they defined in the operative clauses. While this kind of assertions are studied by the deontic logic, and such expressions can be object of reasoning (as in some of the systems described in Section 3), the MCO representation does not aim at matching any specific formal system. The RDF statements of the MCO contract are thus limited to represent the operative clauses in a machine-readable form.

The MPEG-21 MCO standard foresees electronic

contracts directly created in RDF as well as electronic contracts derived from existing ones in natural language. For this reason, being able to declare mappings between the digital and the narrative forms is an important requirement, and for any deontic expression in the digital form it is possible to declare the textual clauses whence it came from.

Contract clauses are identified as well as the contract itself and any other relevant entity. The relationship between contracts in MCO is as important as in any business agreement, and the standard comprises the means for declaring prevalence, cancellations and amendments among other relationships.

The MCO consists of a core model, which provides the elements to make generic deontic statements, and an extension which provides the vocabulary to describe the commercial exchange of exploitation rights of intellectual property assets. More extensions might be envisaged for other kinds of objects, like specific types of content, raw data, etc. For example, as of March 2015, a second edition of the MCO standard has started to be discussed, including a new extension for payments and notifications and a new extension for the expression of rights expression language (MPEG-21 REL) acts [18].

The MCO core ontology consists of 666 OWL2 axioms³ of moderate complexity, adding relatively few semantics over the CEL XML but favoring the publication of contracts as linked data and enabling SPARQL queries over RDF –also integration with other domains is favored.

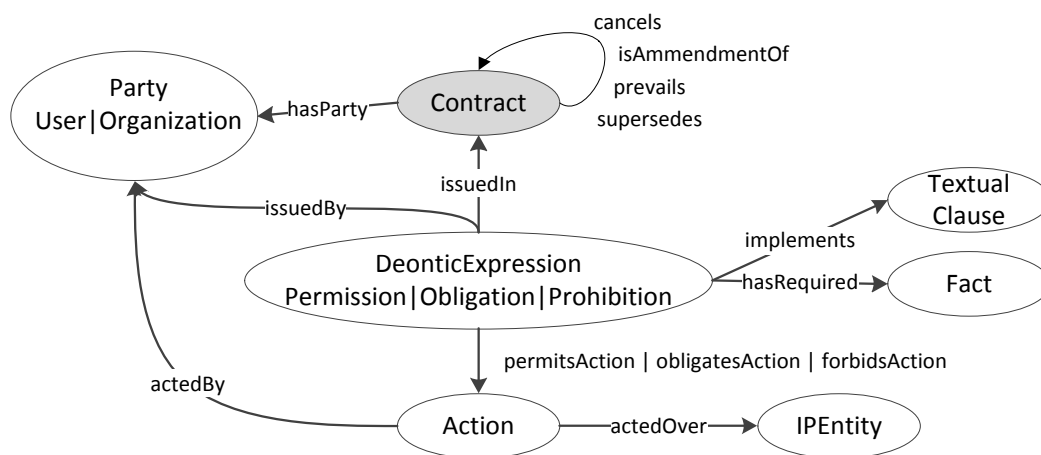


Figure 1. Main classes and object properties of the MCO. Classes are represented with ellipses, object properties with arrows (linking domain and range classes). Prefixes, either mvco or mco-core have been omitted in this figure.

³OWL2 is needed, as some negative object properties have been asserted.

4.2. The MCO Core model

The MCO Core model, whose URI ⁴ is `urn:mpeg:mpeg21:mco:core:2012` and its preferred prefix is `mco-core`, provides the elements for the expression of *permissions*, *obligations* and *prohibitions* – the general terms that a contract usually handles. These statements are the promises that are exchanged in the contract between the *parties*. In fact those patterns are also tackled in various languages for rights expressions, as shown in the comparison made in [19].

Given that MCO is imbricated in the MPEG-21 framework, the description of these parties as well as some of the clauses tying them, is done in terms of elements which are already defined in (or derived from) other parts of the standard, mostly the Media Value Chain Ontology (MVCO [2]). Class instances of the classes in MVCO, whose URI is `http://purl.oclc.org/NET/mvco.owl`, can represent the transformation of one entity of a kind of intellectual property object (an `mvco:IPEntity`) into another one, as well as the required permissions to execute these transformations.

The general overview of the MCO Core model is shown in Figure 1.

4.2.1. Contract and parties

The Media Contract Ontology defines an `mco-core:Contract` class, whose instances represent the actual documents – a contract is thus uniquely identified by an IRI. Contracts can be referred from other contracts to be superseded, altered or extended.

Contracts have parties, which can be natural or legal persons (e.g. people, organizations, etc.), and they are ultimately signed with the digital signature of actual users representing themselves or their organizations, as shown in Figure 2.

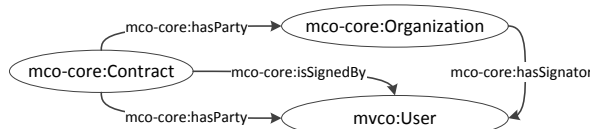


Figure 2. Contracts, organizations and users

Some of these elements are expected to be further

⁴ As MCO is OWL2, identifiers are IRIs instead of URIs (IRI – Internationalized Resource Identifier: a generalization of URI permitting use of Unicode characters), although the standard makes reference to both indistinctly.

described with metadata attributes, i.e. users can get a vCard by using the `hasvCard` property (see [20]) or be attributed with standard Dublin Core elements⁵.

Contracts can be totally or partially encrypted by using the XML Encryption Syntax and Processing [21] and linked to the Contract instance by the `mco-core:encryptedContractPart` object property.

4.2.2. Contract clauses

Any contract, as an exchange of promises, contains a set of deontic expressions: *permissions*, *prohibitions* and *obligations*. The MVCO defined the `mvco:Permission` class, from which the rest could have been expressed—the three deontic modalities can be simplified by using a single one. However, to ease the use of MVCO, specific classes complement the permission: `mco-core:Prohibition` and `mco-core:Obligation`.

The permission model revolves around the concept of *deontic expression* (i.e. a permission, obligation or prohibition) which can be issued by a user (`mvco:User`), and permits an action (`mvco:Action`) having as required zero or more facts to hold (`mvco:Fact`). The action can be further described as being acted by a certain user or set of users over zero or more IP Entities (see Figure 3).

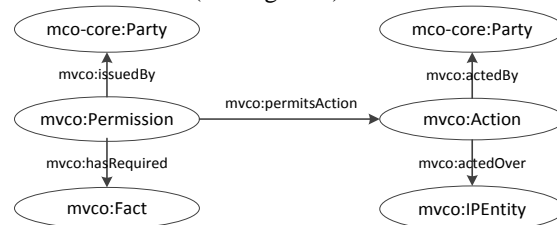


Figure 3. Permission model in MCO

If the deontic expression (`mco-core:DeonticExpression`) is an *Obligation* or a *Prohibition* (`mco-core:Obligation` and `mco-core:Prohibition` respectively), the object property `mvco:permitsAction` is replaced by `mco-core:obligatesAction` and `mco-core:forbidsAction` respectively.

A contract, thus, is in essence a set of *DeonticExpressions*, which implement one or more natural language contract clauses (see Figure 4).

⁵The ISO standard 15836, which establishes a standard for cross-domain resource description, is known as “Dublin Core Metadata Element Set”

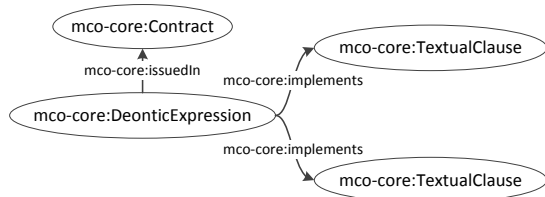


Figure 4. Deontic expressions are issued in contracts and implement textual clauses

4.2.3. Expression of conditions

In standard contracts, permissions are seldom given in exchange of nothing, and conditions are present in almost every contract. To express conditions, the set of facts that must hold is given. However, the actual conditions are not declared by the MCO core language, but by its extension presented in Section 4.3 (or other possible extensions). The MCO uses the class `mvco:Fact` (any sentence with truth value) and the boolean operators to join atomic Facts into more complex facts.

The boolean operators (represented by the class `mco-core:FactComposition`) are `mco-core:FactUnion` (\vee), `mco-core:FactIntersection` (\wedge) and `mco-core:FactNegation` (\neg), following the pattern used in the privacy ontology PPO [22]. The arguments of these logical functions are given by means of the object property `mco-core:hasFact`, as illustrated in Figure 5. No rule (SWRL, SPIN etc.) maintains the consistency of the truth values in the fact compositions, being this task left to contract management software implementers.

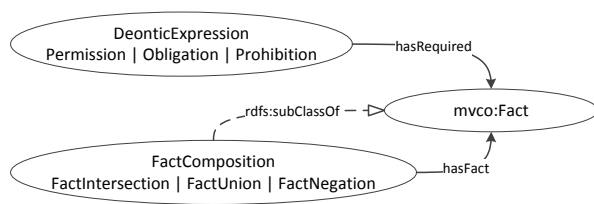


Figure 5 – Boolean operators with Facts. Non prefixed terms are in the `mco-core` namespace. The dashed line represents the `subClassOf` property, going from subject to object. The `DeonticExpression` can be any of its subclasses (`Permission`, `Obligation`, `Prohibition`) as well as the `FactComposition` (which can be `FactIntersection`, `FactUnion` or `FactNegation`).

4.2.4. Example

If Alice wants to allow Bob to make a derivative work of one of her works, called `MyWork`, the minimal set of class instances to be defined would be:

- A class instance representing the contract.
- Two class instances for the parties, representing Alice and Bob.
- A class instance representing Alice’s Work.
- A class instance representing the permission.
- A class instance of the `Action`, in this case instance of the `mvco:MakeAdaptation` class.

Figure 6 shows all the instances along with the main class they belong to.

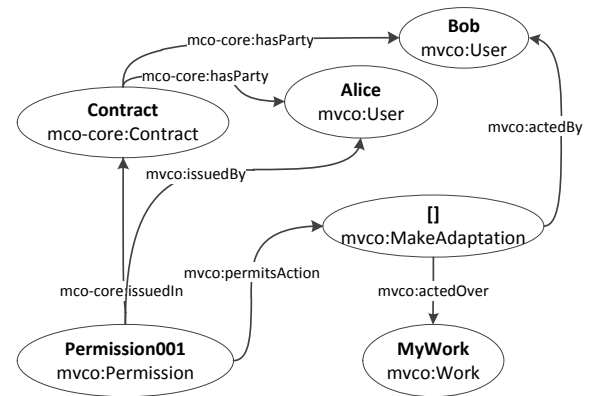


Figure 6. Set of instances to represent a permission from Alice to Bob to make adaptations of her work. Class individuals are represented in boldface (empty brackets meaning ‘blank node’). Arrows are also connecting the class individuals.

The equivalent code for the Figure 6 is shown in Table 1.

Table 1. RDF serialization for Figure 6

```

:Alice a mvco:User.
:Bob a mvco:User.
:MyWork a mvco:Work.
:Contract a mco-core:Contract .
:Contract mco-core:hasParty :Alice, :Bob .
:Permission001 a mvco:Permission;
  mvco:issuedBy :Alice ;
  mco-core:issuedIn :Contract ;
  mvco:permitsAction [
    a mvco:MakeAdaptation;
    mvco:actedBy :Bob ;
    mvco:actedOver :MyWork
  ] .

```

4.3. Extension for Exploitation of Intellectual Property Rights

This extension is intended to represent the most relevant information in media contracts both for con-

tent and services on content based on MPEG-21 technologies.

This is implemented using some specific vocabulary of the MPEG-21 MVCO classes and extensions thereof to represent the main information in media contracts, including specific elements to address the most relevant information found in those contracts for permitting the exploitation of intellectual property rights. Its URI is `urn:mpeg:mpeg21:mco:ipre:2012` and the preferred prefix is `mco-ipre`.

The media contracts on content usually convey permissions to execute one of the generic actions found in `mco-ipre`, like *Duplication* or *Broadcasting*, without going into the particularities of the MPEG-21 REL rights (for example a particular *Enlarge* operation would be better represented with ISO/IEC 21000-5⁶). This MCO extension for Media Contracts is able to represent the most common rights in contracts in the media field and the most frequent conditions in those documents, expressed as Facts which can be required on Permissions or in Deontic Expressions in general.

4.3.1. Description

The MCO extension for exploitation of intellectual property rights does not define any new class without a superclass from `mco-core`, and it fully conforms to the contract model defined by the Media Contract Core.

A number of new classes is defined as a hierarchy of sub-classes of `mvco:Action`, in order to reflect the exploitation actions as defined by the common protection of the intellectual property. The root of such hierarchy is the *Action* named `mco-ipre:ExploitIPRights`, which encompasses all the actions specified by its sub-classes. The defined organization of actions into a hierarchy allows any deontic expression to apply to an action at the desired level of generality/specificity. Granting a general right allows permitting all its specializations. The mentioned hierarchy is depicted in Figure 7.

Another set of classes allows reflecting the various dimensions which form the space of conditions and restrictions. The root of such hierarchy is a sub-class of *Fact* named `mco-ipre:ExploitationCondition`.

Some of the exploitation conditions are expected to be related to other class individuals: `mco-ipre:-AccessPolicy`, `mco-ipre:DeliveryModality`, `mco-ipre:Device`, `mco-ipre:Means`, `mco-ipre:ServiceAccessPolicy`, and `mco-ipre:UserTimeAccess`.

The specification of these terms and others in the MCO-IPRE was not derived from the initial list of requirements but from the experience in contract management of the institutions in the working group.

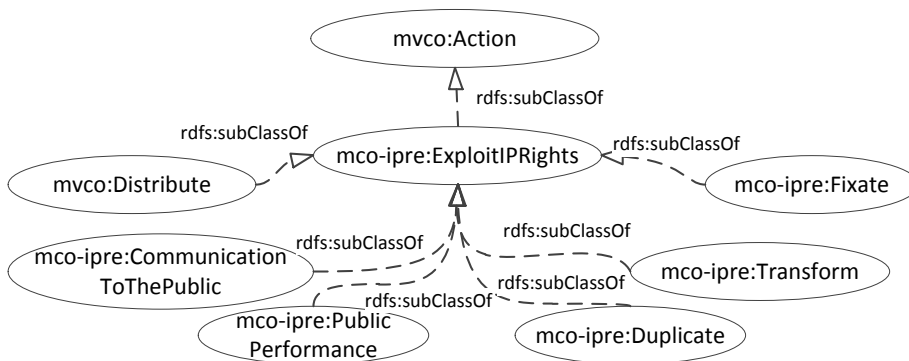


Figure 7 - Diagram of `mvco:Action` hierarchy for exploitation rights. As before, dashed arrows depict subclassing relationships.

⁶ The extensión RELE proposed in [18] covers this aspect.

Other sub-classes of `mco-ipre:ExploitationCondition`, like: `mco-ipre:Language`, `mco-ipre:Length`, `mco-ipre:Run`, `mco-ipre:SpatialContext`, and `mco-ipre:TemporalContext`, are expected to be described with the appropriate datatype properties. The MCO-IPRE extension is indeed completed with the definition of a number of these data properties. Most of them are grouped as sub-properties of the generic data property `mco-core:factProperty` and are defined for allowing a full specification of the exploitation conditions. Finally, other properties, grouped as sub-properties of `mco-core:deonticProperty`, are attributable to a `mvco:Permission`.

It can be observed that many conditions on the exploitation of rights are actually related to the modalities of content fruition by the final users. An example is the hierarchy under `mco-ipre:DeliveryModality`. The legacy communication-to-the-public acted by the broadcasters is “linear”, with a time-schedule over which the viewers have no control. Conversely the modality is “non-linear” if a catalogue is offered to the viewers for independent selection. If the exploitation is constrained over this dimension, one Fact of this hierarchy (i.e. `mco-ipre:Linear`, `mco-ipre:-NonLinear` or one of its sub-classes) has to be required. Another dimension of this kind is `mco-ipre:AccessPolicy`, which deals with the possible payment charged to the final user for the content fruition. If the exploitation requires that no payment can be subsequently asked (except for the governmental fees or taxes related to the public service), the Fact `mco-ipre:FreeOfCharge`, is used. The opposite condition is `mco-ipre:Pay` that can be further specified with `mco-ipre:PayPerView` and `mco-ipre:Subscription`. Two other classes of conditions relate to the modality of making the content available to the final users.

A particular policy of user access (`mco-ipre:ServiceAccessPolicy`) can be constrained to a media service provider by requiring either `mco-ipre:Open`, with which all consumers must be allowed to access the service without need of approval, or `mco-ipre:Restricted`, with which it is required that service provider had explicitly registered the consumers. Notice that a restricted service is not necessarily a pay service.

Eventually, it is possible to require that the time

left to the final user for content fruition is limited or unlimited (by the facts `mco-ipre:Limited` and `mco-ipre:Unlimited` respectively; being both sub-classes of `mco-ipre:UserTimeAccess`). For the former case, the condition can be refined by specifying for how long with the `mco-ipre:hasValidity` data property.

The model also covers the possibility of defining conditions related to the technology used in the delivery to the final user (`mco-ipre:Means`) and related to the type of equipment intended to be used for the fruition by the final user (`mco-ipre:Device`). The delivery channel can be specified to be broadcasting, mobile (broadcast or telecommunications), Internet and other. The device can be specified to be a television set, a computer, a mobile device or others.

The fact `mco-ipre:Length` can be used to constraint the maximum duration of the material actually resulting from the exploitation action. This is typically required when permitting the use of excerpts.

Another condition that was required to be modelled is that of `mco-ipre:IPEntityContext`, that restricts the exploitation action to a specified editorial context. This condition is frequently originated by the will of performers or organizations contributing to a production for keeping the use of the created material limited to the context of the production.

Other exploitation conditions are more self-explaining. The `mco-ipre:SpatialContext` is used for specifying the territory of the exploitation, while `mco-ipre:TemporalContext` gives the license period. The language, e.g. for dubbing or subtitles, can be constrained with `mco-ipre:Language`. Eventually `mco-ipre:Run` is in the domain of a number of data properties, that specify how to take into account the permitted number of executions of an action along time.

Finally, the class `mco-core:ActionRelatedFact`, subclass of `mvco:Fact`, represents the status of accomplishment of the related action, further refined with `mco-core:ActionStarted` and `mco-core:ActionDone`. These classes permit to model the definition of these-called “ancillary-rights”, an example of which is often mentioned in narrative contract text as “Catch-upTV”.

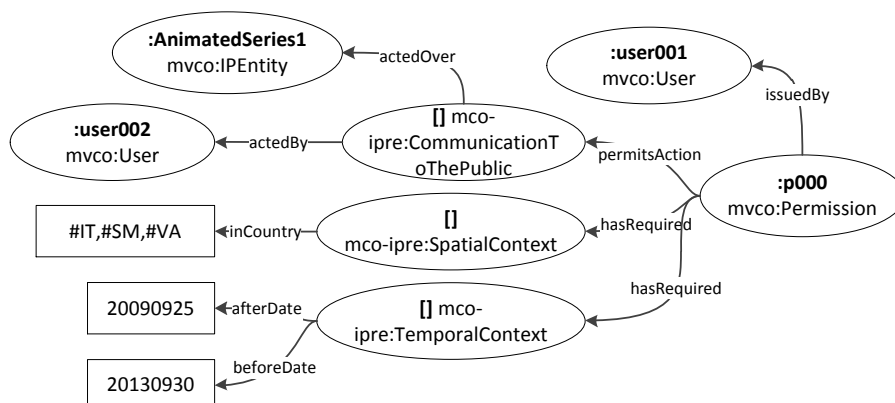


Figure 8. Expression of a permission, extracted from a real contract. Literals are represented in boxes, and ellipses represent class individuals (URI in boldface and main class below, with brackets denoting ‘blank node’). Arrows represent object or datatype properties relating the resources. Prefix has been omitted for the properties.

In such cases, a secondary exploitation action (e.g. make available on the web) is permitted only during a time interval specified by the occurrence of a primary exploitation action (e.g. TV broadcast).

4.3.2. Example

The example in Figure 8 depicts a more complex situation, derived from a real contract, where the permission to make a broadcast is given, subject to certain restrictions (spatial, temporal and of policy of access).

The equivalent code for Figure 8 is given in Table 2.

Table 2. Turtle serialization for Figure 8

```

:user001 a mvco:User.
:user002 a mvco:User.
:AnimatedSeries1 a mvco:IPEntity.
:p000 a mvco:Permission;
  mvco:permitsAction [
    a mco-ipre:CommunicationToThePublic;
    mvco:actedBy :user002;
    mvco:actedOver :AnimatedSeries1
  ];
mvco:issuedBy :user001;
  mvco:hasRequired [
    a mco-ipre:SpatialContext;
    mco:inCountry "#IT;#VA;#SM;"
  ], [
    a mco-ipre:TemporalContext;
    mco-ipre:afterDate "20090925";
    mco-ipre:beforeDate "20130930"
  ], [
    a mco-ipre:FreeOfCharge
  ].

```

5. Practical aspects and examples of use

This section deals first with some practical aspects related to handling MCO contracts. In particular, it discusses the role of the original textual contracts, the specific relationship with content, the procedure to follow with encrypted contracts, and the cases in which an MCO contract document can be binding. Second, a detailed sample contract in the context of audiovisual preservation is examined. Finally, the workflow for creating a MCO contract from an original textual one is also presented and actually followed for the real example.

5.1. Guidance to MCO aspects

5.1.1. Textual contracts and MCO contracts

MCO can be used for representing pre-existing narrative contracts. The process of converting a textual contract into an electronic contract starts with an analysis that identifies the most important terms in the contract and the related deontic elements. Each of the permissions, obligations, and permissions is translated one by one, irrespective of the number of clauses it was spanning in the textual version. Each deontic expression can point to the precise narrative text excerpts which it implements. In addition, the complete textual version can be embedded in the MCO contract.

For new contracts, it is reasonable to directly create them as MCO, with unambiguous and machine-readable deontic expressions. Professional users of the legal domain can verify the semantics by means

of graphical user interfaces of software tools. However these users might be interested in having the possibility to read the equivalent narrative text (a similar approach was described in [23]). Such text can be derived, as the diagrams are, from MCO statements, according to the user's language, and it is not necessary to have it persistently associated to the contract.

5.1.2. Referencing MPEG content

Similarly to the contract entity, also the IP-entities object of the contract can be uniquely identified by their IRI. However, two additional identification mechanisms can be used for referencing content from MCO. First, in the case of MPEG content, a Digital Item Identifier (Part 3 of MPEG-21) can be used as data property of the IP-Entity. Besides, it is possible to annotate the IP-Entity with Dublin Core identifiers for referencing identifications possibly used in other environments and/or legacy systems; for example the ISWC (International Standard Musical Work Code), defined by ISO 15707, or the ISAN (International Standard Audiovisual Number defined by ISO 15706).

5.1.3. Encrypting the contract

In some cases there is the need to keep part or the whole contract encrypted, as it may be considered confidential. For addressing this need, MCO defines a specific data property of the contract class itself, that is named `mco-core:encryptedContractPart` and has as range a `XMLLiteral`.

The unencrypted version of the contract can be obtained by decrypting the value of the data property and merging the result with the unencrypted part of the contract. In practice, this process is almost straightforward in the case of OWL/XML or RDF/XML serialization of the MCO contract. Conversion from/to XML or other serialization, if desired, must be implemented at both ends.

The suggested practical algorithm for encrypting the MCO contract is the following:

- Serialize the contract with XML syntax.
- Identify contract elements for which encryption is requested, and extract them into a separate document.
- Encrypt the contract elements required and insert the result as value of the contract data property in the unencrypted part.

To obtain the unencrypted version of the contract,

the value of the data property should be unencrypted and merged with the rest of the unencrypted contract.

5.1.4. Contract templates

For a contract to be binding, it requires the signatures of all parties. If this is not the case, it means that the contract is falling in one of the possible situations in which it is not finally agreed. For instance, one party might have defined an offer that is ready for being proposed to potential parties, who might accept it "as is" or begin a negotiation with subsequent modifications before final common approval.

Another case is that of templates which are used by large organizations or in specific contexts as the basis of rights negotiations and trades. In this context, the parties know very well the terms already defined in the templates and they only have to discuss and agree on the differences, that will include identification of the other party, the object of the contract, and usually the conditions related to spatial and temporal contexts and other details. Deontic expressions defined in the template but not applicable to the particular case will be simply removed.

5.2. Examples of use

5.2.1. MCO for audiovisual preservation

The long term audiovisual preservation scenario, which is the scope of PrestoPRIME [5][24] and Presto4U [25] projects, is based on future re-use of archive content for fruition by future users also with commercial exploitation by the archive holders as a mean for funding the preservation process itself. This scenario requires precision of knowledge on the owned rights related to the archival items, and on the rights origination as well. For a particular item, the rights owned by the content holder may derive from several (sometimes many) contracts and it must be possible:

- to have strong relationship between rights and content, which may imply to have the rights as part of the archival information package;
- to check a target exploitation for the content with a single contract or with all the related owned rights.

5.2.2. Creating an MCO Contract from an existing one in paper

We assume that it is possible to have a scanned electronic version of the paper contract, which might

be suitable for use by a human editor of the MCO contract. However it will be even better to have a text document of it, which has to be carefully verified if obtained by OCR (optical character recognition)

techniques. A real example of contract, from which sensible or confidential information has been removed, is presented in Table 3.

Table 3. Example of an original paper contract

PROGRAMME LICENCE
XXXX (licensor) and RAI (licensee)

THIS AGREEMENT (the "Agreement") is made as of June the day 30 of 2008 BETWEEN
(1) XXXXX, [...], ("XXXX" / Licensor); and
(2) RAI [...], Italy (the Licensee).
collectively, the Parties

PREMISES
XXXX and the Licensee agree that the Licensee shall be licensed to transmit the Programmes (as hereinafter defined).
In consideration for one another's promises and with immediate effect, the Parties to this Agreement agree as follows:

Programme: [...title of program...]

For avoidance of doubt, Licensee acknowledges that XXXX owns all rights in the dubs.

Language: Italian
Licensed rights: Free Television rights, via air, cable and satellite; by analogue and digital technologies and whether encrypted and/or un-encoded.

For purposes of this Agreement, transmission and/or re-transmission by means of cable and by satellite (both in the analog and digital format) and whether encrypted and/or unencoded, shall be deemed Free Television to the extent that it constitutes a simultaneous technical extension of the free terrestrial broadcast signal, by any technologies and/or protocols, by wire/cable or wireless, so that the original audio/video signal may be carried on telecommunication platforms of any nature and kind, on both fixed and mobile networks, including those ones to be implemented and developed in the future, and intended for reception by any kind and typologies of terminals/devices.

For avoidance of doubt, it is herewith expressly stated that Free Terrestrial Television rights - licensed under this Agreement - include exploitation by both analogue and digital terrestrial transmission.

Territory: Italy, Vatican City, San Marino and Monte Carlo, including simultaneous retransmission.

Licence Period: 4 years from [...]

Number of Tx: 4 (Four)

For avoidance of doubt, anyhow broadcast/transmitted/distributed/disseminated, the repeated diffusion/broadcast of the Programme in the 24 hours following its first diffusion/broadcast shall be, for the purposes of this Agreement, equal to one television run.

The licence under this Agreement is granted on an exclusive basis.

In order to safeguard the exclusivity of the hereby licensed rights, XXXX undertakes and agrees that, in the event such rights are available to XXXX itself, it shall not license to any third party in return for no payment from such third party, in the Territory, in the Language and during the Licensed Period/s, the right to communicate to the public any of the Programme object of this Agreement by any and all free of charge forms of circular diffusion/broadcast (point to multipoint), on any platforms and by any technical means and/or technologies and/or communication protocols accessible/receivable/viewable by any type of terminals/devices, including the multimedia ones, both fixed and mobile.

Exclusivity: The licence under this Agreement is granted on an exclusive basis.

Special Conditions:
[...]
For the avoidance of doubt nothing in these Special Conditions or the main Terms of Business shall entitle the Licensee to either subtitle the Programmes or broadcast them in any language other than the Language herein granted.

IN WITNESS whereof the hands of the parties or their duly authorised attorneys or representatives the day and year first above written.

Agreed and accepted for and on behalf of XXXX
By.....Signed.....
Title.....

Agreed and accepted for and on behalf of RAI S.p.A.
By.....
Signed.....

The process for contract creation from a text contract described hereinafter is the one followed by

using the Rightsdraw editing service [26] described in Section 6.2. Figure 9 shows the contract creation workflow.

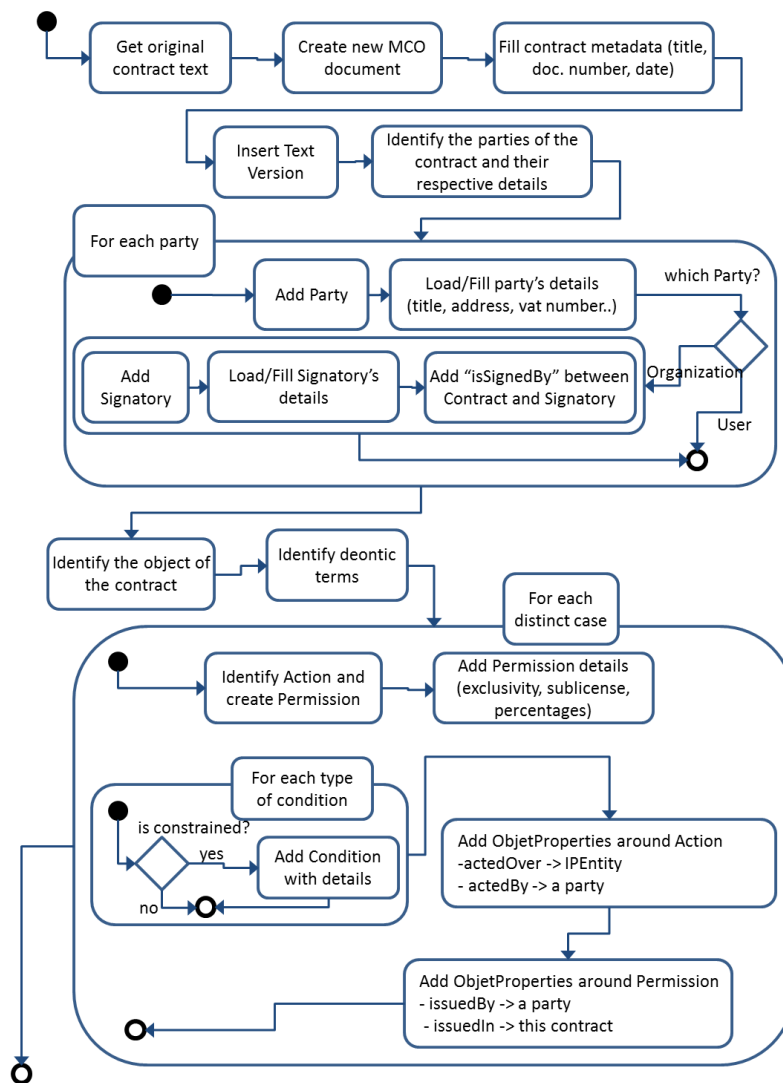


Figure 9. Contract creation workflow from an original contract

Having the text of the original contract, the process starts by creating a new MCO document with its own IRI, importing the appropriate extension of MCO, such as `mco-ipre`, indicating all the required prefixes, and containing a *NamedIndividual* of class `mco-core:Contract`, which will represent the contract itself.

The contract is then identified by its IRI. Dublin Core annotation can be used for establishing a title and for setting the contract date. The narrative textual version of the contract can be inserted immediately as a data property of the contract individual.

The second step consists of identifying the parties of the contract, and inserting them into the MCO document with all the desired details, with the help of Dublin Core annotations. Each party is identified by its IRI, but other information can be found in the narrative text that can be inserted in the MCO document. In this example we use `dc:title` for the name of the organization, `dc:identifier` for providing its international VAT (Value Added Tax) Number, and the data property `mco-core:Address`.

If the party is an Organization, as in this case, it will be necessary to indicate a signatory User, in order to let the contract to be binding.

The third step introduces Intellectual Property Entities according to its definition in the Media Value Chain Ontology permission model.

Thus, the person analyzing the narrative contract will have to identify in the text the parts that originate the permissions. This can be a difficult task in the case of complex narrative contracts, because a preliminary sentence that might indicate a single permission can be later developed in different cases in which the particular conditions are defined, so that several distinct permissions have to be created.

The subsequent step is to ensure that all the object properties are properly declared. In particular, for each permission it has to be known exactly: in which contract it is issued, which party is the issuer, and

which party can act the permitted action; eventually, but mostly important, over which object (content or service) the action can be acted. A similar procedure applies to obligations and prohibitions.

Optionally, it is possible to indicate for each deontic expression which is the part of narrative contract text that is implemented, in order to provide evidence about the validity of the mapping process from the original paper contract.

The purpose of a last step for validation is twofold. On the one hand, it is required to ensure that all terms have been covered. On the other hand, it is required that each deontic expression is faithfully represented. For this task, a service deriving a synthetic narrative text from the MCO contract can be very useful. Figure 10 and Figure 11 show diagrams of the MCO document in the various editing steps.

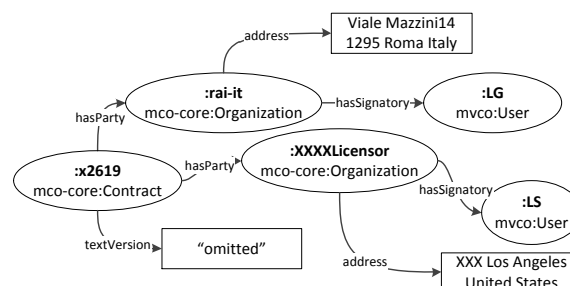


Figure 10. Diagram representing the initial steps of creating a MCO contract document

The signatories indicated above were not identified in the original available contract sample, although the placeholders were explicitly provided.

Once with the contract represented as a MCO RDF document, it is possible to parse the entities and generate natural text from it. Table 4 presents the text possibly derived by parsing the MCO contract entities, while Table 5 provides the RDF serialization of the same MCO contract.

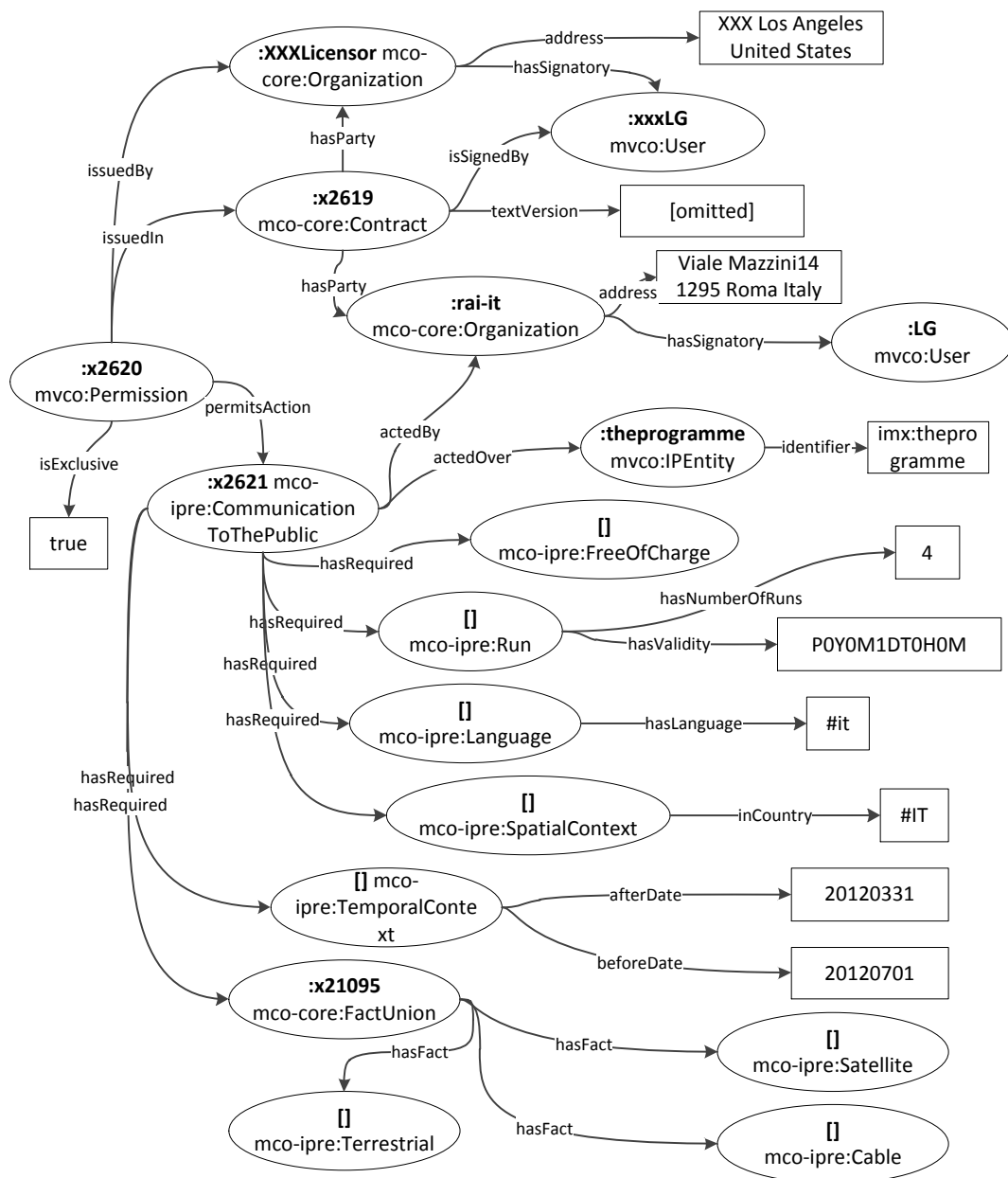


Figure 11. Diagram representing the permission resulting contract document

Table 4. Contract text derived by parsing the MCO contract

<p>MCO Contract: [#x2619] TITLE: PROGRAMMELICENCE DATE: 2008-06-30</p> <p>Parties The present agreement is between:</p> <p>rai.it title: RAI - RadiotelevisioneItalianaS.p.A.</p>
--

identifier: VATIN:IT06382641006
Address: viale Mazzini 14\n001295 Roma\nItaly

Represented by: LG

XXXX-Licensor

title: XXXX Company
Address: xxxx\nLos Angeles\nUnited States

Represented by:XXXX-LS

Relationships with other Contracts
None

Object(s) of this agreement

IPEntity: theprogramme
Identifier: [imx:theprogramme]
title: title of the programme

Agreed Deontics

Permission: #x2620

Grants to rai.it, on an exclusive basis, with right of sublicense, the 100% (one hundred per cent) of use and exploitation, and the 100% (one hundred per cent) of income receipts the permission to:

CommunicationToThePublic (#x2621) - acted on: **the programme**, which is the resource identified by: imx:theprogramme, provided that all of the following Facts hold:

- The access policy is **FreeOfCharge**, that is no payment is required for access, within the execution of the action.
- The exploitation is constrained to **Run**, that is the action can be executed at most 4 times. It is agreed that a single run shall mean the execution of the action within the P0Y0M1DT0H0M validity time period.
- The exploitation is constrained to **Language**, that is the language of IP-Entity object of the action must be within the following list of language codes: #it;
- The exploitation is constrained to **SpatialContext**, that is the action can be executed only within the territory identified by the following country codes: #VA;#IT;#SM;#MC;
- The exploitation is constrained to **TemporalContext**, that is the action can be executed only within the time period identified as follows:
after: 20080831
before: 20120901

- Case: **#x21035** - It is required that **at least one** of the following Facts hold:
 - The specific technology used for the transmission of audiovisual content to end users is restricted to be **Cable**, that is the means are restricted to be any Broadcast Technology which makes use of co-axial and/or fibre optic cable for direct reception by a TelevisionSet. It does not include DSL or other Internet or Ip-based networks.
 - The specific technology used for the transmission of audiovisual content to end users is restricted to be **Satellite**, that is the means are restricted to be any Broadcast Technology which makes use of a geostationary satellite system.
 - The specific technology used for the transmission of audiovisual content to end users is restricted to be **Terrestrial**, that is the means are restricted to be any Broadcast Technology which makes use of a terrestrial television transmitter.

Signatories

LG

XXXX-LS

Table 5. Turtle serialization for the created MCO contract

@prefix : <urn:it.rai:mco-rights-mco.2013-09-22.2618.owl#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .


```

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@base <urn:it.rai:mco-rights-mco.2013-09-22.2618.owl> .

# First, the parties are declared
:LGrdf:type mvco:User , owl:NamedIndividual .
:XXXX-LS a mvco:User , owl:NamedIndividual .
:XXXX-Licensors a owl:NamedIndividual, mco-core:Organization ;
    dct:language "English"^^xsd:string ;
    dct:title "XXXXCompany"^^xsd:string ;
    mco-core:Address "xxxxLosAngelesUnitedStates"^^xsd:string ;
    mco-core:hasSignatory :XXXX-LS .
:rai.it a owl:NamedIndividual , mco-core:Organization ;
    dct:title "RAI-RadiotelevisioneItalianaS.p.A."^^xsd:string;
    dct:identifier "VATIN:IT06382641006"^^xsd:string ;
    mco-core:Address "vialeMazzini14\\n001295Roma\\nItaly"^^xsd:string ;
    mco-core:hasSignatory :LG .

# Second, the asset is declared
:theprogramme a mvco:IPEntity , owl:NamedIndividual ;
    <urn:mpeg:mpeg21:2002:01-DII-NS#Identifier> "imx:theprogramme"^^xsd:string ;
    dct:title "title of the programme"^^xsd:string .

# Third, the contract is instantiated
:x2619 a owl:NamedIndividual , mco-core:Contract ;
    mco-core:TextVersion " (TEXT REMOVED - FULL TEXT CONTRACT HERE)"
    dct:date "2008-06-30"^^xsd:dateTime ;
    dct:title "PROGRAMMELICENCE"^^xsd:string ;
    mco-core:isSignedBy :LG , :XXXX-LS ;
    mco-core:hasParty :XXXX-Licensors , :rai.it .

# Permission granted in the contract
:x2620 a mvco:Permission , owl:NamedIndividual ;
    mco-ipre:isExclusive "true"^^xsd:boolean ;
    mvco:issuedBy :XXXX-Licensors ;
    mco-core:issuedIn :x2619 ;
    mvco:permitsAction :x2621 ;
    mvco:hasRequired [
        a mco-ipre:SpatialContext ;
        mco-ipre:inCountry "#VA;#IT;#SM;#MC;"^^xsd:string] ,
    [
        a mco-ipre:Run ;
        mco-ipre:hasNumberOfRuns "4"^^xsd:decimal ;
        mco-ipre:hasValidity "P0Y0M1DT0H0M"^^xsd:duration] ,
    [
        a mco-ipre:Language ;
        mco-ipre:hasLanguage "#it;"^^xsd:string ] ,
    [
        a mco-ipre:FreeOfCharge] ,
    [
        a mco-core:FactUnion ;
        mco-core:hasFact [
            a mco-ipre:Terrestrial] ,
        [
            a mco-ipre:Cable] ,
        [
            a mco-ipre:Satellite]],
    [
        a mco-ipre:TemporalContext ;
        mco-ipre:afterDate "20080831"^^xsd:date;
        mco-ipre:beforeDate "20120901"^^xsd:date
    ] .

# Action that is granted
:x2621 a owl:NamedIndividual , mco-ipre:CommunicationToThePublic ;

```

```
mvco:actedBy :rai.it ;  
mvco:actedOver :theprogramme .
```

6. Developments

The reference software of an MPEG standard is an implementation which demonstrates the use of the standard. This section describes first some software modules that are now part of the MCO and CEL reference software (those modules are very much related since the Contract Expression Language was developed in parallel with MCO, as explained in section 1). Then, a complete system for creation and management of MCO contracts (Rightsdraw) is also presented.

6.1. CEL and MCO reference software

This section describes the software developed for media contracts management, focusing on preservation of digital rights and intellectual property protection. First, it presents the modules specifically implemented for the creation, search and permission based authorization of MCO contracts. Then, it presents the modules developed for the identification, storage, validation, authorization, search and delivery of CEL (Contract Expression Language) contracts. Finally, the integration of MCO and CEL modules is presented taking advantage of the MCO-CEL converter, which justifies the use of CEL software in a MCO environment.

6.1.1. Native MCO software modules

The native MCO reference software modules have been implemented by RAI, and are derived from a version of the Rightsdraw software presented below in Section 5.2. The main module implements the basic CRUD (create, read, update, delete) functionalities. The user of the software can work through an editing interface in which the MCO document is presented by a diagram from which a number of HTML forms are served to the user for active interaction. Another module implements the “*check with*” and “*search*” functionalities described in the requirements document [7].

“*Check-with*” creates a query, in terms of target exploitation and compares it with the indexed Permissions returning the list of MCO documents having Permissions matching the target exploitation. The Permission does not need to be exactly the same, but must be compatible with the target one, i.e. it must have less or equal conditions, or conditions with wider boundaries, and same or more general permitted actions.

The example of contract presented in Figures 10 and 11 was created by using the editing modules of the MCO reference software. The same example is used here for describing the *Check-with* interface and presenting the result of its use. The application interface is shown in Figure 12.

MCO-Reference Software

Username: Access

Welcome default to your Query service page.

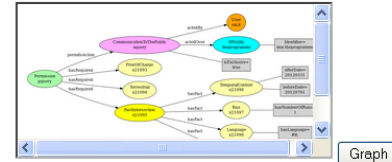
Query which archival items match with the Permission that you will define

Do Check with for: exclusive:

User: IPEntity: #theprogram identifier:

Conditions	negative flag	Conditions	negative flag
Free of Charge	<input type="checkbox"/>	after	<input type="checkbox"/>
ServiceAccessPolicy: any	<input type="checkbox"/>	before	<input type="checkbox"/>
DeliveryModality: any	<input type="checkbox"/>	country codes	<input type="checkbox"/>
Terrestrial	<input type="checkbox"/>	language codes	<input type="checkbox"/>
Device: any	<input type="checkbox"/>	number of runs	<input type="text" value="1"/> - / +
UserTimeAccess: any	<input type="checkbox"/>	run validity	<input type="text" value="null"/>
		number of reps	<input type="text" value="null"/> - / +

Your query



Found 1 items matching your query



Figure 12. MCO Reference Software: form for Check-with activity with example of input and its result

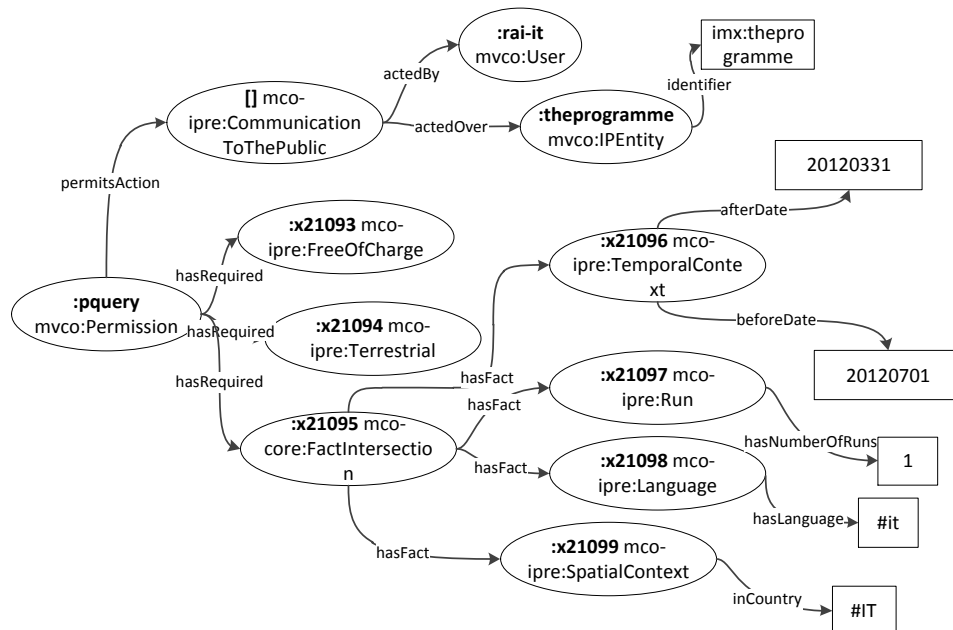


Figure 13. Diagram representing the target permission used in the Check-with example

The query form is filled with values matching the conditions required in the contract example: the target “Means” is one out of the three options required in the *FactUnion*; the time of context of the exploitation falls in the contract license period, as well as the territory; the number of target runs is lower than the

maximum allowed; the two other conditions as “FreeOfCharge” and “Language” match exactly, as well as the User, the *IPEntity*, and its digital item identifier. Figure 13 shows in more detail the diagram of the *target permission* used in this query example

The MCO reference software uses OWL/XML serialization as default, however it can import and export from/to RDF/XML serialization as well. Moreover, it integrates the MCO-CEL converter module (see Section 5.1.3), allowing import of contracts expressed in CEL and export of MCO contracts as CEL documents.

6.1.2. CEL software modules

The CEL reference Java software modules include:

- *Contract Identification*: allows a user to obtain a unique identifier for a contract.
- *Contract Storage*: stores a contract expressed in MPEG-21 CEL format.
- *Contract Validation*: validates syntactically an MPEG-21 CEL contract, determining if the contract is valid or not and the reasons why.
- *Contract Delivery*: allows a Service Provider to deliver a contract to one user
- *Contract Search*: searches contracts according to a set of terms and conditions provided by a user. The search can be performed by matching the parameters *Text* in the contract, *Contract Identifier*, *Party* and *IPEntity* against the contract to which they apply. The input query format is defined in MPEG Query Format (ISO/IEC 15938-12) [27].
- *Contract Check-with*: verifies if a requested action against a contract matches with its content. The result of the operation includes one of these values: (i) `OK`, if the action can be performed; (ii) `USER_NOT_FOUND`, if the service invoker is different from one of the contract parties; (iii) `DEONTIC_EXPRESSION_NOT_FOUND`, if a deontic expression is missing or incomplete; (iv) `CONDITION_NOT_SATISFIED`, if the deontic expression is present but there is at least one condition that is not satisfied or (v) `UNKNOWN_ERROR`, if any unknown error occurs.

The modules Contract Identification, Contract Storage, Contract Validation and Contract Delivery share a common structure, having a contract as input and a response to the corresponding operation, as shown in Figure 14. The response indicates if the operation is successful or not.

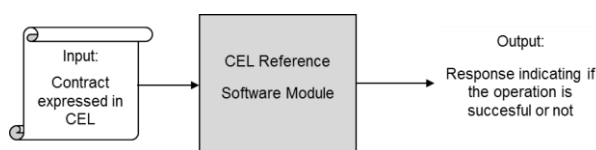


Figure 14. Common structure for Contract Identification, Storage, Validation and Delivery CEL reference software modules

The *Contract Search* module does not receive a contract as input, but a query expressed in MPEG Query Format. The result of the operation is the response to the query. The structure of the module is very similar to the one in Figure 14 as it only receives an input and provides an output.

Finally, the Contract Check-with module has a slightly different structure, as shown in Figure 15. In this case, the inputs are a contract, an authorization context, the action to be authorized and the content over which the action will be performed. The output of this module is the result of the contract based authorization.

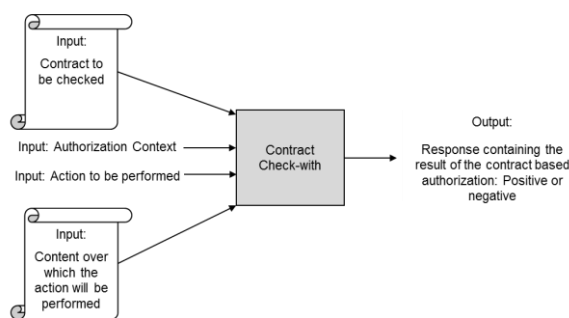


Figure 15. Structure of Contract Check-with CEL reference software module

6.1.3. MCO-CEL converter

As the work in MPEG-21 around CEL and MCO was carried out in the same time span and in a collaborative environment, both support the same contract elements and concepts. This entails the possibility to select between the two of them, depending on criteria related to the technical environment and context of use. Besides, it is possible to conceive services for conversion from CEL to MCO and vice-versa, on limited and tested scenarios.

Since not all the reference software modules have been implemented for both CEL and MCO in the same environment, the use of this module permits the conversion between the two expressions of the contract in order to apply the corresponding reference software module. For instance, if we want to perform the Contract Check-with operation over a contract expressed in MCO, we could first convert from MCO to CEL and then use the result of this operation as an input for the CEL's Contract Check-with module. This way of working can be applied to all the reference software modules implemented for CEL and

MCO when the input contract is not expressed in the supported language.

Figure 16 shows the structure of the software modules implemented for both conversions.

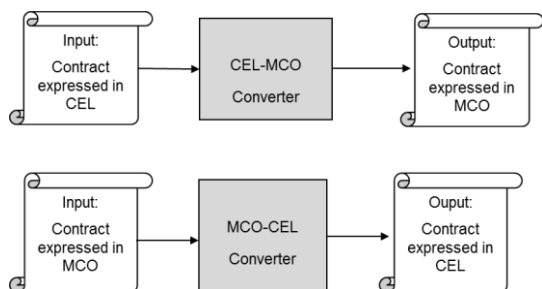


Figure 16. Structure of MCO-CEL converter reference software module

The conversion is performed differently in both directions. In the rest of the sub-section we briefly describe the workflow of CEL-MCO conversion and MCO-CEL conversion.

The CEL to MCO conversion involves the following steps:

1. A CEL contract is parsed from an input file.
2. CEL parties are parsed and converted into MCO ontology elements. The parsing is aware if the party is a Person or an Organization and creates the proper properties for the party.
3. Contract metadata is parsed. Contract identifier and parties' references are considered.
4. The clauses' metadata and its elements are parsed. Regarding the constraints, only fact intersections, unions and negations are considered in order to create the hierarchical constraint structure.
5. An MCO contract containing the contract elements is returned (namely, instances of the MCO ontology and their relations).

The MCO to CEL conversion is done in the following steps:

1. The RDF MCO contract is parsed and a memory model generated.
2. Contract identifier is set in the CEL contract.
3. MCO parties are converted to CEL format. The parsing is aware if the party is a Person or an Organization and creates the proper related class to the party.
4. The clauses are converted. The following clause elements are considered (i) Issuer; (ii) Object; (iii) Subject and (iv) Resultant Object. Constraints and acts are not considered

5. The XML CEL contract is generated from the memory model.

6.1.4. How to use MCO and CEL reference software together

The MCO and CEL reference software can be considered all together as a larger set of modules, using the MCO-CEL converter as pivot.

Assuming that a specific contract will be finalized either in CEL or in MCO, the conversion between the two formats can be helpful during the contract lifecycle, such as for editing or validation, or afterwards for *Check-with* operations or for delivery to systems operating in the other format. The MCO-CEL conversion is therefore a critical component, as it bridges simple XML applications with semantically enabled tools.

As a use case example, an operator who has just edited our contract example by using the MCO reference software editing modules is informed that the contract must also be delivered to a system expecting CEL documents. The form for exporting the document also accepts CEL as input, in addition to OWL serialization options.

The implementation of the converter module is limited to the main elements. The MCO party entities are mapped to CEL using the IRIs as IDs of the CEL party elements and the related Dublin Core annotation assertions are also found and mapped. The MCO Permission goes into the `operativePart` element of CEL, as a `deonticStructuredClause` element, having the ID derived from the Permission IRI, in which the two MCO triples (`Action, actedBy, Organization`) and (`Action, actedOver, IPEntity`) are mapped into CEL by means of the elements: `subject`, `act`, and `object`. The required facts are only partially supported by the converter.

6.2. Rightsdraw

Rightsdraw is a set of services for creating and handling MCO documents, available in the web [28], from which the MCO reference software was mostly derived. Till end of 2012 it has been developed in the framework of European funded project PrestoPRIME, as a proof of concept rights management system [26].

Each user of the service can create and work on MCO documents stored on her repository. The editing activity can be done through the diagram which is itself an interaction interface for modifying the MCO document, by means of the served user forms. The *Check-with* service can be used for finding Permis-

sions matching a target exploitation. It is also possible to update rights holding information, i.e. the set of Permissions hold over a certain IP-Entity, with the application of a new contract, implementing thus purchases and sales scenarios.

Rightsdraw also supports the definition and use of “key patterns”, which are templates of recurrent permissions, the use of which is partially shown in Fig-

ure 17: the permission related to a content item is checked against the defined patterns, recognized by a simple title, without considering conditions specified by data properties. The user of the service can see those details of each permission, obtain a derived text, delete a permission, and add a new one based on the selected pattern. The patterns themselves can be edited using the diagram based interface.

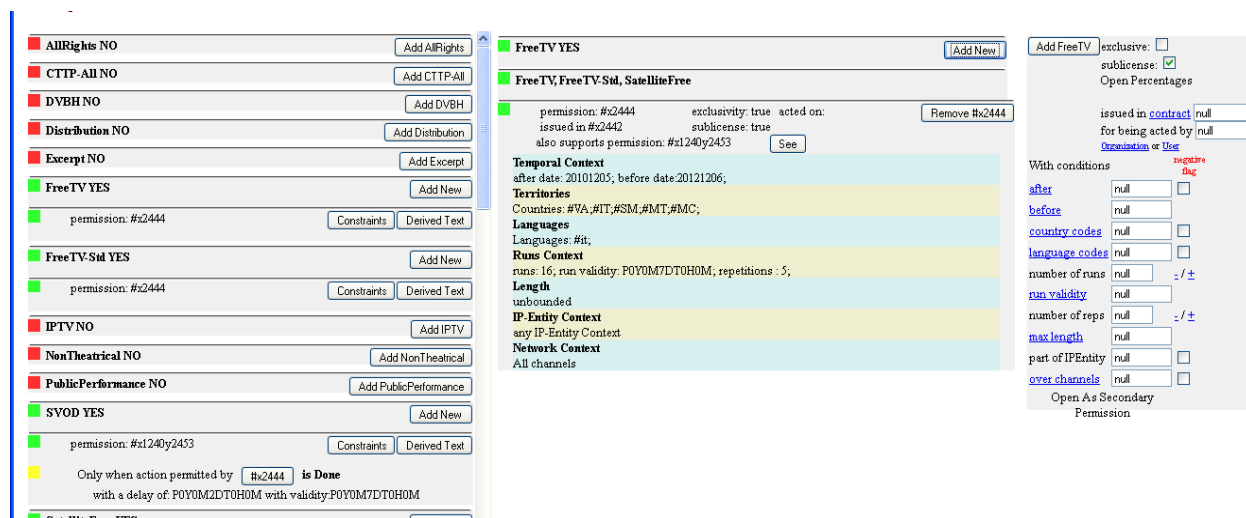


Figure 17 - Rightsdraw interface for editing rights on the basis of user defined patterns

7. Conclusions and future work

The MPEG Standard committee has been working on the specification of two ways of representing contracts, which have been finalized as two new parts of MPEG-21, namely CEL (ISO/IEC 21000-20) and MCO (ISO/IEC 21000-21). The paper describes the contract model and key elements in MCO, such as the parties in the contract and the relevant clauses conveying permissions, obligations and prohibitions.

Other initiatives have tried to represent contracts, as described in Section 2. However, MCO addresses the needs related to the deal of media rights, based on the legal framework on the protection of the intellectual properties and characterized by complex multi-dimensional conditions: to identify the main contract entities and have a machine-readable operative part to support the media industry operations. One example is that of rights clearance work, which is related to planning of production, broadcast events, and archive repurposing. Another case is that of purchase and sale of media rights.

Besides, other metadata standards and initiatives, such as MPEG-7 [29] or EBU-Core [30], are established regarding the media content description to support the content publication and content search and retrieval. Such formats have found big difficulties in covering rights aspects, because of the complexity of the rights domain, with its need to resolve the truth of any evaluation of rights, and typically these formats leave placeholders for undetermined rights statements or plain text (not machine readable). The MCO can cover this gap or missing link, in RDF, as the content description standards can point to the IRIs of media contracts or even to the deontic expressions related to their content of interest, while from the rights management environment the IP-Entities can have links to their descriptive information.

The other main contributions of the paper are related to the creation process of a contract (to be represented in MCO) and the tools needed to achieve that. The different examples provided, all of them verified and implemented, confirm the usefulness of MCO and how to use it.

The authors of this paper have been the main contributors to the specification of the standard and have

developed the tools (and Reference Software) described. Future work on this topic includes improving the software tools by rationalizing the contract process creation. In fact, the MCO standard has already been corrected (a first “corrigendum” has been approved [31]) to fix some minor issues discovered during software development.

Another line for future research revolves around the ODRL (Open Digital Rights Language) that is a well-established standard for representing rights expressions, whose version 1.1 issued in 2002 [32] was based in XML. The ODRL 2.1 specification [33] defines a OWL ontology for its Core Model, much in line with the Media Contract Ontology, with the intention of representing the rights expressions also with RDF statements. The alignment of the MCO and the ODRL 2.1 Core Model ontologies looks as a logical next step towards gaining interoperability and better acceptance in the Linked Data community.

Acknowledgements

Part of this work has been funded by the Spanish project *Protección, búsqueda e interoperabilidad de contenidos multimedia: Nuevas técnicas y aplicaciones (PBIInt)* (TEC2011-22989), by the European Commission in the seventh framework program (PrestoPRIME, FP7-ICT-2007-3 231161 and Presto4U, ICT 2011.4.3 Coordination Action GA No. 600845) and a Juan de la Cierva research contract by the Spanish Ministry of Science and Innovation. The authors thank Annarita Di Carlo (RAI / Teche) for providing the contract example and verifying the mapping.

References

[1] ISO/IEC 21000-21:2013. Information Technology —Multimedia Framework (MPEG-21) —Part 21: Media Contract Ontology, 2013

[2] ISO/IEC 21000-19:2010. Information Technology —Multimedia Framework (MPEG-21) —Part 19: Media Value Chain Ontology, 2010

[3] ISO/IEC 21000-20:2013. Information Technology —Multimedia Framework (MPEG-21) —Part 20: Contract Expression Language, 2013

[4] Rodríguez, E., Delgado J., Boch, L., Rodríguez-Doncel, V. Media Contracts Formalization Using a Standardized Contract Expression Language, IEEE Multimedia, doi:10.1109/MMUL.2014.22, April 2014

[5] PrestoPRIME, EU project – Keeping Audiovisual Content Alive – Available at <http://www.prestoprime.eu>, accessed Nov. 2013

[6] Di Carlo, A., Boch, L., Lucaferri, L., Messina, A., Picciotti, G., Allasia, W., Gallo, F., Lanza, S., Todarello, E., Delgado, J.,

Rodríguez, E., Rodríguez, V., Chiariglione, F., “Common Rights Ontology”. PrestoPRIME Deliverable ID4.0.5B, available at https://prestoprimews.ina.fr/public/deliverables/PP_WP4_ID_4.0.5b_RightsOntology_R0_v1.11.pdf, accessed Nov. 2013

[7] ISO/IECJTC 1/SC 29/WG11 Systems Group, N11228 - Requirements for Advanced IPTV Terminal (AIT), Jan. 2010

[8] ISO/IECJTC 1/SC 29/WG 11 Systems Group, N11386, "Core Experiment for Contracts mapping", July 2010

[9] Rubino, R., Rotolo, A., Sartor, G., 2006. An OWL Ontology of Fundamental Legal Concepts., in: van Engers, T.M. (Ed.), JURIX, Frontiers in Artificial Intelligence and Applications. IOS Press, pp. 101–110.

[10] Hoekstra, R., Breuker, J., Bello, M.D., Boer, A., 2007. The LKIF Core Ontology of Basic Legal Concepts, in: Casanovas, P., Biasiotti, M.A., Francesconi, E., Sagri, M.T. (Eds.), Proceedings of the Workshop on Legal Ontologies and Artificial Intelligence Techniques (LOAIT 2007).

[11] Guth, S., Strembeck, M. (2004). A Proposal for the Evolution of the ODRL Information Model. In Proc. of the ODRL Workshop, pp. 87-106.

[12] Delgado, J., Rodríguez, E., Rodríguez-Doncel, V., Baldinato, M., Bixio, F. Use of MPEG-21 REL for the expression of Audiovisual Contracts. ISO/IECJTC1/SC29/WG11M15430. April 2008.

[13] Governatori, G. and Milosevic, Z. A formal analysis of a business contract language. International Journal of Cooperative Information Systems, 15(4):659–685, 2006

[14] Prisacariu, C., Schneider, G. A dynamic deontic logic for complex contracts. Journal of Logic and Algebraic Programming 81 (4), 458-490, 2012.

[15] Kobryn, C., Atkinson, C., Milosevic, Z., Electronic Contracting with COSMOS - How to Establish, Negotiate and Execute Electronic Contracts on the Internet. In: 2nd Int. Enterprise Distributed Object Computing Workshop (EDOC '98)

[16] Yan, J. Zhang, Y., Yan, M. "Ontology Modeling for Contract: Using OWL to Express Semantic Relations". Enterprise Distributed Object Computing Conference, 2006. EDOC '06. 10th IEEE International, pp.409-412, 2006

[17] Leff, L., & Meyer, P. (2007). eContracts. Version 1.0. Technical report, OASIS. <http://docs.oasis-open.org/legalxmlecontracts>.

[18] ISO/IECJTC 1/SC 29/WG 11 Systems Group, N15194, "Text of ISO/IEC CD 2nd edition 21000-21 Media Contract Ontology (MCO)", February 2015

[19] Rodríguez, V., Suarez, M., Gomez, A. Poveda, M. Licensing patterns for Linked Data. In Proc. of the 4th Int. Workshop on Ontology Patterns, 2013.

[20] Ianella, R., McKinney, J. (eds.): vCard Ontology, W3C Working Draft 24 September 2013

[21] Eastlake, D., and Reagle, J. (eds), "XML Encryption Syntax and Processing". W3C Recommendation (2002)

[22] Sacco, O., Passant, A., "A Privacy Preference Ontology (PPO) for Linked Data". In Proc. of the Linked Data on the Web, LDOW 2011

[23] Rodríguez, V., Delgado, J., and Rodríguez, E.: "From Narrative Contracts to Electronic Licenses: A Guided Translation Process for the Case of Audiovisual Content Management". In Proc. of 3rd Int. Conf. on Automated Production of Cross Media Content for Multi-Channel Distribution, 2007

[24] Boch, L., Di Carlo, A., Gallo, F., "Model, Format and Services for Audiovisual Rights Management". In Proc. of the 1st Int. Conf. on Inf. Tech. for Performing Arts, Media Access and Entertainment, Firenze Univ. Press (2012)

- [25] Presto4UEU project- European Technology for Digital Audiovisual Media Preservation – Available at <http://presto4.eu>, accessed Nov. 2013
- [26] Boch, L., Di Carlo, A., “Proof of Concept Rights Management System”.PrestoPRIME Deliverable D4.0.6, available at https://prestoprimews.ina.fr/public/deliverables/PP_WP4_D4.0.6ProofOfConceptRightsManagementSystemv1.11.pdf
- [27] ISO/IEC IS 15938:12. Information technology — Multimedia content description interface — Part 12: Query format (2008)
- [28] RightsDraw2, “Technical Report”. Available at http://www.crit.rai.it/EN/attivita/opensource/Rightsdraw_en_2013.pdf, accessed Nov. 2013
- [29] ISO/IEC 15938. Information Technology —Multimedia content description interface
- [30] EBU Tech 3293 version 1.5 - A metadata specification of the European Broadcasting Union (EBU) http://tech.ebu.ch/docs/tech/tech3293v1_5.pdf
- [31] Rodriguez, V., Boch, L., Delgado, J. Material for a first Corrigendum for ISO/IEC 21000-21, ISO/IECJTC1/SC29/WG11 MPEG2013m31429
- [32] Ianella, R. (ed.): Open Digital Rights Language v.1.1. Technical Report, W3C, 19 September 2002
- [33] Ianella, R., Guth,S., Kasten, A.(eds.). ODRL2.0 Core Model Specification, W3C Community Group Specification.