INTERNATIONAL STANDARD

ISO/IEC 21000-20

Second edition
2016-12-15

Information technology — Multimedia framework (MPEG-21) —

Part 20:
Contract Expression Language

Technologies de l'information — Cadre multimédia (MPEG-21) —

Partie 20: Langage d'expression des contrats

Reference number
ISO/IEC 21000-20:2016(E)

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see the following URL: www.iso.org/iso/foreword.html.

The committee responsible for this document is ISO/IEC JTC 1, *Information Technology*, Subcommittee SC 21, *Coding of audio, picture, multimedia and hypermedia information*.

This second edition cancels and replaces the first edition (ISO/IEC 21000-20:2013), which has been technically revised with the following changes:

— some strengthening of the core contract part and some reordering within the extension for the exploitation of intellectual property rights have been provided;

— the set of facts available for specifying conditions has been completed;

— addition of a simple payment and notification extension;

— the relationship with acts defined in ISO/IEC 21000-5 (REL) has been refined;

— a clear mechanism for defining further future extensions is in place.

A list of parts in the ISO/IEC 21000 series can be found on the ISO website.

Introduction

Today, many elements exist to build an infrastructure for the delivery and consumption of multimedia content. There was, however, no "big picture" to describe how these elements, either in existence or under development, relate to each other. The aim for the ISO/IEC 21000 series has been to describe how these various elements fit together. New standards, as appropriate, will be developed while other relevant standards may be developed by other bodies.

The result is an open framework for multimedia delivery and consumption, with both the content creator and content consumer as focal points. This open framework provides content creators and service providers with equal opportunities in the ISO/IEC 21000 series-enabled open market. This will also be to the benefit of the content consumer providing them access to a large variety of content in an interoperable manner. The vision for ISO/IEC 21000 is to define a multimedia framework to enable transparent and augmented use of multimedia resources across a wide range of networks and devices used by different communities.

ISO/IEC 21000 series aims thus at defining an open framework for multimedia applications, where users distribute, consume, operate on and transact with content represented as Digital Items.

These transactions can be automatically governed by licenses using the Rights Expression Language from ISO/IEC 21000. However, beyond the operative information present in a digital license, the digital representation of the complete business agreements between the parties may prove useful for a number of purposes. The Contract Expression Language (CEL) is the ISO/IEC language to express such contracts in a structured representation.

CEL may be used to represent contracts, for content directly, or for services on content based on MPEG-21 technologies. However, CEL may also be used as electronic format for contracts on the trade of media rights also beyond the MPEG framework.

The provided features include the identification of the contract itself and of its parties, and an unambiguous expression of the agreed permissions, obligations, and prohibitions, in a machine readable way, so that their verification can be implemented in software.

In particular, the CEL deontic expressions address the rights for the exploitation of intellectual property entities, including the specification of the associated conditions, together with other contractual aspects, such as payments, notifications or material delivery.

However, CEL may also be used as electronic format for contracts on the trade of media rights also beyond the MPEG framework.

The main aspect of CEL contracts are the operative clauses, represented as machine readable deontic expressions, i.e. the agreed permissions, obligations, and prohibitions, and the associated terms and conditions.

Besides, the CEL contract includes the identification of the contract itself, its parties, and the possible relationships with other contracts.

Among the provided features, there is the possibility to insert the textual version of the contract and/or of specific clauses, in particular for the case in which the original contract is narrative, i.e. written in natural language. Also, it is possible to add metadata related to any contract entity and to have encryption of the whole contract, or any sub-part of it. As electronic format for a contract document, the agreement of the parties can be proved by their digital signature.

Eventually, CEL provides to the media companies the basic means for the collection of knowledge on held rights, also derived from multiple contracts, as rights port-folio, for business management purposes.

Various potential benefits can be associated to the use of CEL. Firstly, CEL can support the business of media companies, for product placement and maximizing reuse of archive content, implying also cost reductions in all rights related activities, e.g. rights clearance. Afterwards, it supports the respect of copyright laws with respect to new exploitation technologies, also by contributing to the reduction of mistakes with respect to contract compliance, implying decreasing number of controversies and other cost reductions. In general, CEL aims at increasing the quality of rights information, which gets more reliable and can be integrated with other metadata in standard way. This can bring even to improve working conditions and all the decision processes within media companies.

Information technology — Multimedia framework (MPEG-21) —

Part 20: Contract Expression Language

1 Scope

This document specifies a language for representing contracts in the Multimedia Framework formed for the transaction of MPEG-21 Digital Items or services related to the MPEG-21 Framework.

Contract Expression Language (CEL) aims at digitally representing the agreements made in environments using ISO/IEC 21000. These agreements are contracts for transactions of content packed as Digital Items, as well as services provided around this content.

The range of contracts under scope are as follows:

— contracts about transactions on rights for the exploitation of content as MPEG-21 Digital Items;

— contracts about the provision of MPEG-21-based services, i.e. delivery, identification, encryption, search and others.

The aspects represented by CEL contracts include the following:

— the textual clauses, in natural language as they are in the narrative contract, duly structured;

— the operative clauses, as computer language expressions.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 21000-5, *Information technology — Multimedia framework (MPEG-21) — Part 5: Rights Expression Language*

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— IEC Electropedia: available at http://www.electropedia.org/

— ISO Online browsing platform: available at http://www.iso.org/obp

3.1.1
action
something done or performed by a *party* (3.1.6) or anyone acting on her behalf

Note 1 to entry: Action also represents the exercise of a right or duty.

3.1.2
deontic expression
generic entity encompassing the properties of an agreed machine readable contract clause regulating the *actions* (3.1.1) of the *parties* (3.1.6)

3.1.3
MPEG-21 Contract
representation of agreements formed for the transaction of MPEG-21 Digital Items or services related to the MPEG-21 Framework

3.1.4
MPEG-21 Service
system supplying utility in the MPEG-21 Framework

3.1.5
obligation
deontic expression (3.1.2) binding one of the *parties* (3.1.6) to execute an *action* (3.1.1)

3.1.6
party
organization or a user who accepts to respect the *deontic expressions* (3.1.2) defined in the contract and who expects the other *parties* (3.1.6) in the same contract to do the same

3.1.7
permission
deontic expression (3.1.2) allowing one of the *parties* (3.1.6) to execute an *action* (3.1.1)

3.1.8
prohibition
deontic expression (3.1.2) binding one of the *parties* (3.1.6) not to execute an *action* (3.1.1)

3.2 Abbreviated terms

CEL	Contract Expression Language
IPRE	Intellectual Property Rights Exploitation
IRI	Internationalized Resource Identifier
MCO	Media Contract Ontology
MPEG	Moving Picture Experts Group
MPEG-7	ISO/IEC 15938
MPEG-21	ISO/IEC 21000
OWL	Web Ontology Language
RDF	Resource Description Framework
REL	Rights Expression Language
URI	Uniform Resource Identifier (IETF Standard is RFC 3986)
URN	Uniform Resource Name (IETF Standard is RFC 2141)
W3C	World Wide Web Consortium
XML	Extensible Markup Language (W3C Recommendation)

4 Conventions

4.1 Document conventions

XML Representation

The syntax of each XML element in the Contract Expression Language is specified using the constructs provided by XML Schema. XML Schema documents or its fragments are presented in orange boxes. Omissions are marked with suspension points ([...]).

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" [...]
```

XML documents or its fragments are presented in gray boxes. Omissions are marked with suspension points ([...]).

```
<cel-core:Contract id="715" xsi:schemaLocation="urn:mpeg:mpeg21:cel:core:2015
cel-core.xsd" xmlns:dc="http://purl.org/dc/elements/1.1/" [...]
```

This document also makes use of diagrams to express portions of XML Schema, as shown in Figure 1.

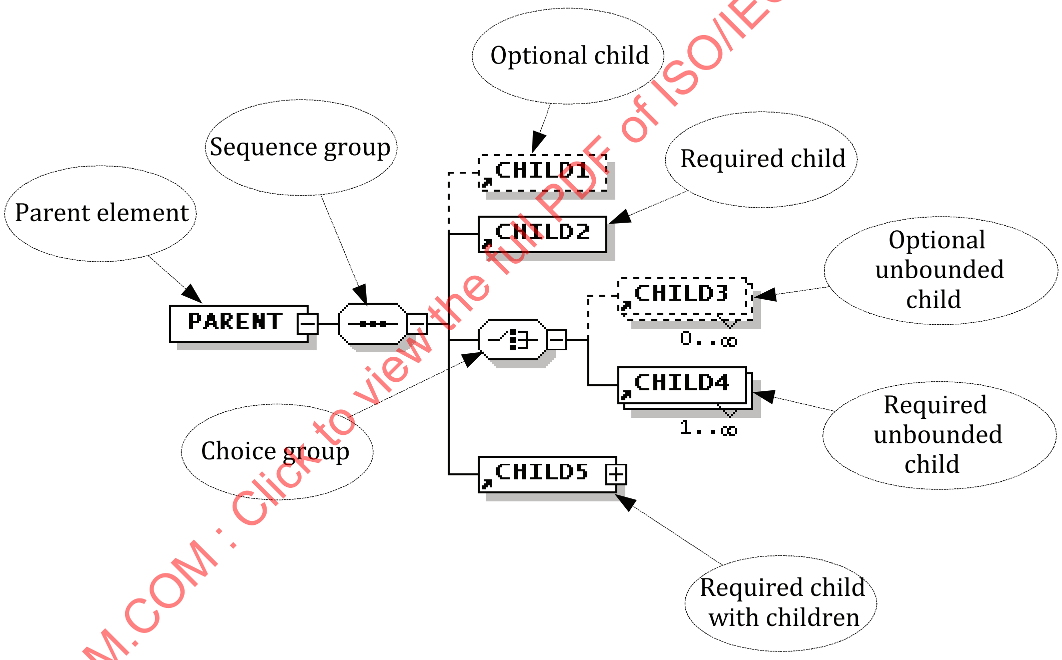

Figure 1 — Sample XML Schema diagram

Figure 1 states which elements are required (boxes with solid outline), those that are optional (boxes with dashed outline), the number of occurrences of each element (0...), and the lineage between elements (symbols between elements indicating either a choice, or a sequence).

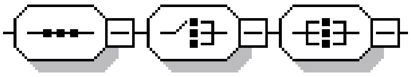

Figure 2 — Compositors used in XML Schema diagrams

Compositors, as shown in Figure 2, describe (from left to right), respectively, a *sequence* of elements, a *choice* of elements and the *all* model.

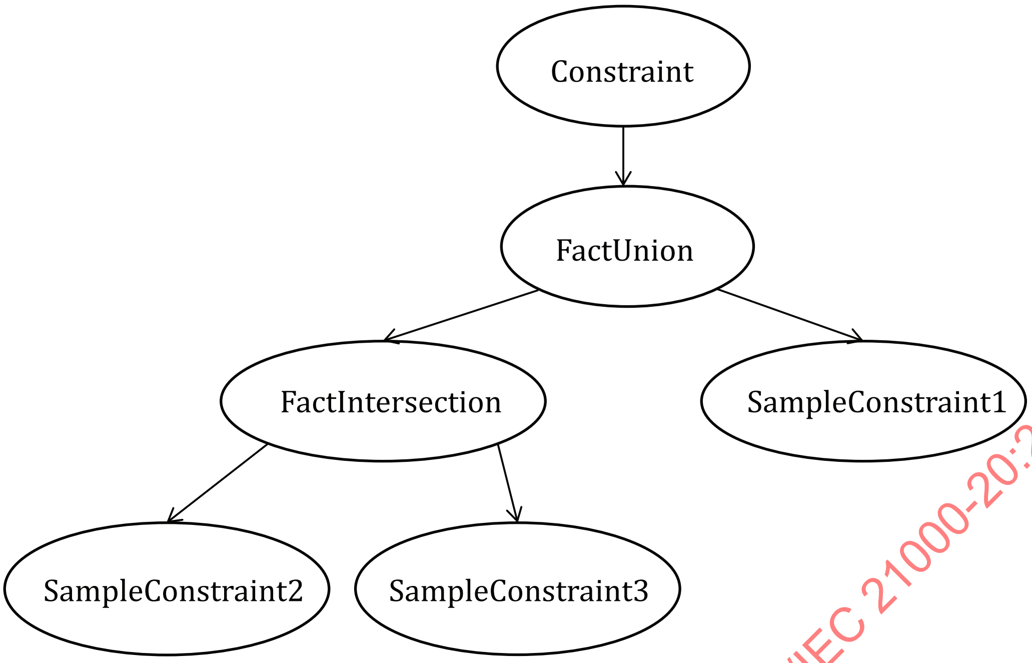

Figure 3 — Graphical representation of contract examples

Graphical description of contract examples, as shown in Figure 3, presents how the elements in a contract relate between themselves to facilitate understanding of complex representations. FactUnion and FactIntersection allow the combination of constraints to represent complex relationships between the constraints defined in this document to represent a contract.

4.2 Namespace prefix conventions

4.2.1 General

The namespace for CEL core XML Schema is:

```
urn:mpeg:mpeg21:cel:core:2015
```

The namespace for CEL extension on exploitation of intellectual property rights XML Schema is:

```
urn:mpeg:mpeg21:cel:ipre:2015
```

The namespace for CEL extension on payments and notification XML Schema is:

```
urn:mpeg:mpeg21:cel:pane:2015
```

The namespace for CEL extension on MPEG-21 REL standard and multimedia extensions is:

```
urn:mpeg:mpeg21:cel:rele:2015
```

4.2.2 Use of prefixes

The CEL makes use of elements defined in other schemas, either given by MPEG-21 or others. The used namespace prefixes together with their reference is given in Table 1.

Table 1 — Mapping of prefixes to namespaces in examples and text

Prefix	Corresponding namespace	Ref
cel-core	urn:mpeg:mpeg21:cel:core:2015	Here
cel-ipre	urn:mpeg:mpeg21:cel:ipre:2015	Here
cel-pane	urn:mpeg:mpeg21:cel:pane:2015	Here
cel-rele	urn:mpeg:mpeg21:cel:rele:2015	Here
dc	http://purl.org/dc/elements/1.1/	ISO 15836
dii	urn:mpeg:mpeg21:2002:01-DII-NS	ISO/IEC 21000-3
dsig	http://www.w3.org/2000/09/xmldsig#	W3C XML Signature Syntax and Processing
rel-mx	urn:mpeg:mpeg21:2003:01-REL-MX-NS	ISO/IEC 21000-5
rel-r	urn:mpeg:mpeg21:2003:01-REL-R-NS	ISO/IEC 21000-5
rel-sx	urn:mpeg:mpeg21:2003:01-REL-SX-NS	ISO/IEC 21000-5
xenc	http://www.w3.org/2001/04/xmlenc#	W3C XML Encryption Syntax and Processing
xsd	http://www.w3.org/2001/XMLSchema	W3C XML Schema
xsi	http://www.w3.org/2001/XMLSchema-instance	W3C XML Schema Instance

5 Relationship to other parts of ISO/IEC 21000

The Digital Item is the fundamental unit of distribution and transaction in the Multimedia Framework. While the different parts of ISO/IEC 21000 deal with the components and different aspects of Digital Items, together they form a complete integrated interoperable framework. This clause describes the relationship of this document with the other parts of ISO/IEC 21000 in addressing the representation of the agreements for the aforementioned transactions.

A contract represented following this document may become a part of a digital item (whose declaration is given with the ISO/IEC 21000-2). If so, it will be declared with the Type element of ISO/IEC 21000-3 pointing to the CEL URI, as described in B.3.

This document has the aim, as ISO/IEC 21000-21, of digitally representing contracts information. CEL enables the structured representation of contracts' information making use of XML, while MCO provides their semantic representation by means of OWL or RDF. Both parts share the goal of defining a contract document able to provide the information listed in 6.1 and they share the semantics for actions, facts and services.

This document also supports the expression of rights and conditions as defined in ISO/IEC 21000-5.

6 Overview

6.1 General aspects

A Contract Expression Language (CEL) contract is a document providing the following information, optional unless otherwise stated:

— identification of the contract itself — Required;

— possible relationships with other contracts;

— the Parties — Required;

— the textual version of the contract;

— a number of textual clauses which can be referenced by the operative part element;

— the Object of the contract (Content or Service) — Required;

— the Operative part, containing the contract information which have to be machine readable (deontic expressions, links to textual clauses) — Required.

CEL supports the possibility to encrypt either the whole contract or part of it.

CEL aims at providing the structural elements to syntactically represent operative clauses, in a machine-readable form.

6.2 Syntactic representation

A Contract document defined by CEL shall be compliant to the XML representation defined in Clause 7. The document format is specified by an XML Schema. Any representation of a contract document defined by CEL supports the possibility of encrypting either the whole contract or part of it.

Contract documents shall validate against the CEL XML Schema, which is provided in Annex A. A detailed description of the structured representation is given in Clause 7.

7 Contract Structured Representation

7.1 General

CEL standard specification enables the structured representation of digital media contracts by means of XML. To this end, the CEL core XML schema has been defined. It normatively defines the core elements for media contracts. Its URI is:

```
urn:mpeg:mpeg21:cel:core:2015
```

Media contracts consist of deontic expressions which permit, obligate or prohibit users to execute generic actions over digital media if the imposed conditions are fulfilled. Common actions and conditions in media contracts have been defined as acts and constraints in the XML CEL extension on exploitation of intellectual property rights. Its URI is:

```
urn:mpeg:mpeg21:cel:ipre:2015
```

Moreover, it is needed to represent the basic information about payment terms agreed in media contracts and about agreed notification actions intended to inform the contract parties, or other indicated users, about the occurrence of relevant events. This information is defined in the XML CEL extension on payments and notification. Its URI is:

```
urn:mpeg:mpeg21:cel:pane:2015
```

Finally, to facilitate reference to MPEG-21 Rights Expression Language Standard and Multimedia extensions, a separated schema is provided. Contracts not making use of actions or constraints defined in this schema, do not need to include all of them. Its URI is:

```
urn:mpeg:mpeg21:cel:rele:2015
```

MPEG-21 CEL contract documents shall validate against the XML Schema specification(s) described in this subclause.

7.2 XML Schema definition

7.2.1 Contract element and main structure

The root element of a CEL contract is the `cel-core:Contract` element, which includes the structured representation of the contract by means of deontic clauses, as well as the original text version of the contract. It also provides mechanisms for relating the narrative clauses of the contract with its structured XML representation.

The structure of the `cel-core:Contract` element is depicted in Figure 4. Its child elements are as follows:

- optionally, a choice of `cel-core:TextVersion` containing the whole narrative contract as plain text or `cel-core:EncryptedTextVersion` containing an encrypted version of the whole narrative contract;
- `cel-core:Metadata`, optionally including metadata such as contract author, contract language, etc.;
- `cel-core:ContractsRelated`, optionally including references to pre-existing contracts the validity of which might be affected by the present one;
- `cel-core:Party`, the number of parties is open while 0 means that the contract is a template;
- `cel-core:Body`, exactly one body element is mandatory, with the contract clauses;
- `cel-core:EncryptedContract`, a full version of an encrypted contract; if it is present, the rest of elements do not appear.

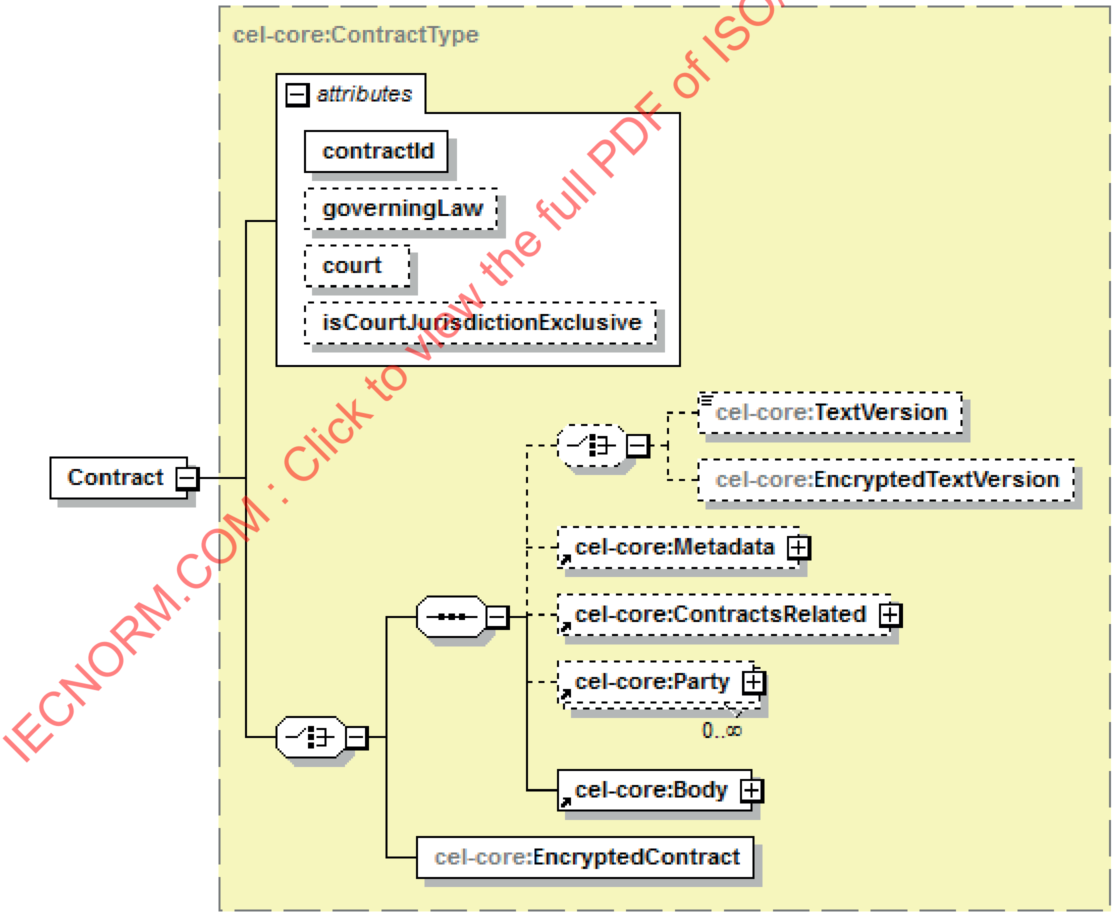

Figure 4 — Root and main elements in the contract

The `cel-core:Contract` element has the following attributes:

- `contractId` which uniquely identifies the contract;

- `governingLaw` for indicating the applicable governing law, i.e. the agreed reference legal system of the contract; it can be indicated as a country, if that country has only one legal system;
- `court` indicating which court has jurisdiction over any dispute related to the terms and conditions defined in the contract;
- `isCourtJurisdictionExclusive` indicating if the jurisdiction of the court, indicated by the attribute court, is agreed to be exclusive (true), preventing the parties from bringing proceedings in another court, or not (false). This attribute should be present if and only if the attribute court is present.

The CEL core XML Schema has the following header.

```
<schema xmlns=http://www. w3 .org/2001/XMLSchema
   xmlns:cel-core="urn:mpeg:mpeg21:cel:core:2015"
   xmlns:dsig=http://www .w3 .org/2000/09/xmldsig#
   xmlns:dii="urn:mpeg:mpeg21:2002:01-DII-NS"
   xmlns:xenc=http://www.w3 .org/2001/04/xmlenc#
   xmlns:rel-r="urn:mpeg:mpeg21:2003:01-REL-R-NS"
   xmlns:dc=http://purl.org/dc/elements/1.1/
   targetNamespace="urn:mpeg:mpeg21:cel:core:2015"
   elementFormDefault="qualified" attributeFormDefault="unqualified"
   version="2nd Edition">
   <import namespace=http://purl.org/dc/elements/1.1/
schemaLocation="http://dublincore.org/schemas/xmls/qdc/2008/02/11/dc.xsd"/>
   <import namespace="urn:mpeg:mpeg21:2003:01-REL-R-NS"
schemaLocation="http://standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-21_schema_
files/rel-r/rel-r.xsd"/>
   <import namespace=http://www.w3.org/2000/09/xmldsig#
schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/xmldsig-core-schema.
xsd"/>
   <import namespace="urn:mpeg:mpeg21:2002:01-DII-NS"
schemaLocation="http://standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-21_schema_
files/dii/dii.xsd"/>
   <import namespace=http://www.w3.org/2001/04/xmlenc#
schemaLocation="http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/xenc-schema.xsd"/>
```

7.2.2 Example

A sample contract may reference the XML Schema this way.

```
<?xml version="1.0" encoding="UTF-8"?>
<cel-core:Contract
   xmlns:cel-core="urn:mpeg:mpeg21:cel:core:2015"
   xmlns:dc=http://purl.org/dc/elements/1.1/
   xmlns:xsi=http://www .w3.org/2001/XMLSchema-instance
   xmlns:dsig="http://www .w3.org/2000/09/
   xmldsig#"xmlns:dii="urn:mpeg:mpeg21:2002:01-DII-NS"
   xmlns:xenc=http://www.w3.org/2001/04/xmlenc#
   xmlns:rel-r="urn:mpeg:mpeg21:2003:01-REL-R-NS"
   xsi:schemaLocation="urn:mpeg:mpeg21:cel:core:2015 cel-core.xsd>
```

7.2.3 Metadata

Metadata can be added under the `cel-core:Metadata` element, giving information about the contract itself. Their optional elements can be seen in Figure 5. A Dublin Core placeholder is provided (element `cel-core:SimpleDC` of type `dc:elementContainer`). Other metadata schemas are supported by means of the element `cel-core:Other`.

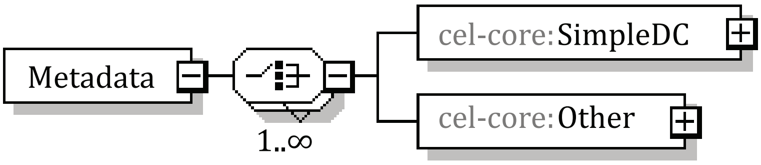

Figure 5 — Contract metadata elements

7.2.4 Element to relate contracts

The element `cel-core:ContractsRelated` provides the means to relate the contract to other identified contracts, between the same partners, as shown in Figure 6. The validity of the related contract can be affected according to relationship type, which shall be one of the following:

- `supersedes`, the referenced contract has to be considered terminated by the new agreement which totally replaces it;
- `cancels`, the parties agree to cancel all the effects of the referenced contract;
- `prevailsOver`, the referenced contract is generally still valid, but in case of conflict the terms of the new one prevail;
- `isAmendmentOf`, the referenced contract is partially modified by the new agreement.

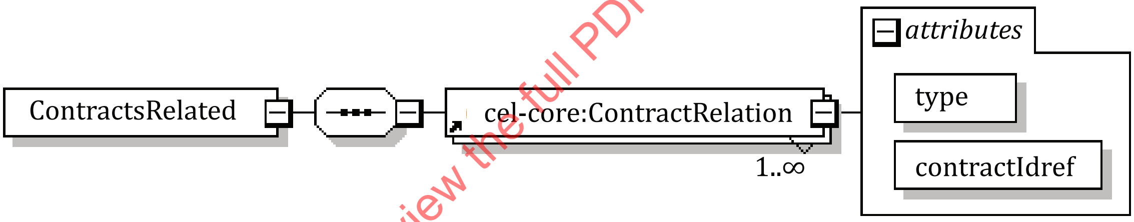

Figure 6 — ContractsRelated element

7.2.5 The parties

7.2.5.1 General

The contract contains the parties in the contract, as shown in Figure 4.

- `cel-core:Party`, zero or more parties for which the contract is binding.

A contract template has zero parties, or placeholder parties with temporary information, while a contract offer has at least one signing party.

Each party, as described in Figure 7, can either be a person (element `cel-core:Person`) or an organization (element `cel-core:Organization`). Both share the same structure (`cel-core:PartyBasicGroup`), but in the latter case, a signatory person can be optionally given (element `cel-core:Signatory`).

The `cel-core:PartyBasicGroup`, common to both Person and Organization cases, and to the signatory element as well, is made of the following:

- `cel-core:Name` element, the name of the party;
- a number of `dc:identifier` elements, to be used for giving references to registries in which the party is registered, such as the VAT identification number;

— the `dc:description` element, to provide a free text description of the party;

— the `cel-core:Details` element, to provide further detailed information, such as address and other contact information.

The party element might be further enriched with the address (element `cel-core:Address`) that can be used to provide party address as free text.

Without the `dsig:Signature` element, the contract is not binding.

The attribute `id`, of type `xsd:ID`, is used for uniquely identify the party element within the contract document.

The `cel-core:PartyType` complex type has been created with the structure depicted in Figure 7.

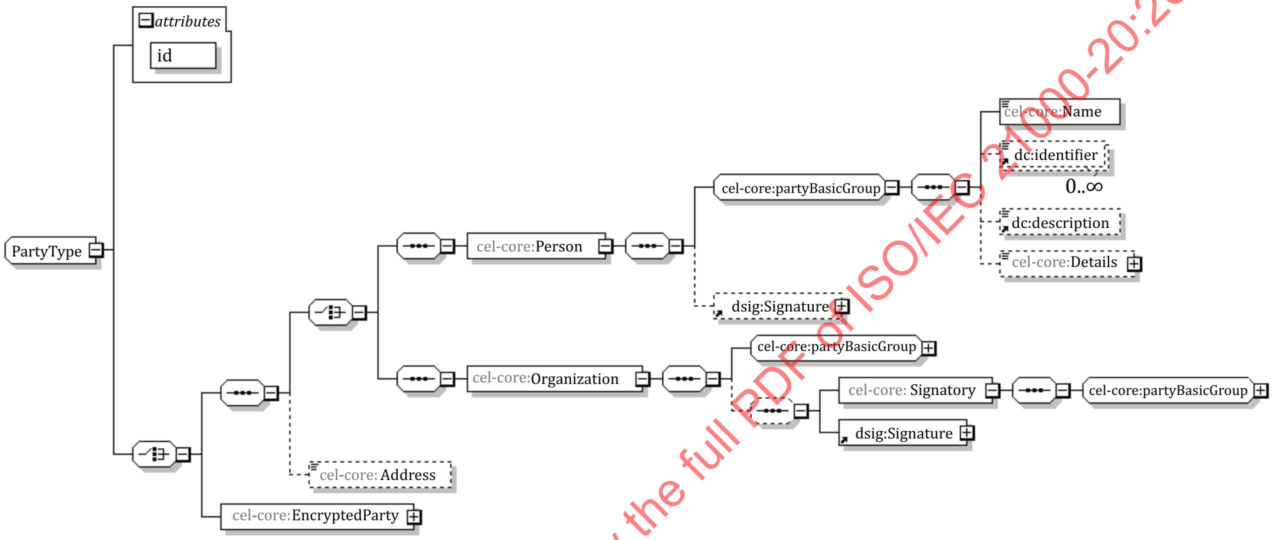

Figure 7 — Contract party element

The relevant elements of the XML Schema used for defining the `cel-core:PartyType` are given in the following box.

```
<complexType name="PartyType">
   <choice>
      <sequence>
         <choice>
            <sequence>
               <element name="Person">
                  <complexType>
                     <sequence>
                        <group ref="cel-core:partyBasicGroup"/>
                        <element ref="dsig:Signature" minOccurs="0"/>
                     </sequence>
                  </complexType>
               </element>
            </sequence>
            <sequence>
               <element name="Organization">
                  <complexType>
                     <sequence>
                        <group ref="cel-core:partyBasicGroup"/>
                        <sequence minOccurs="0">
                           <element name="Signatory" minOccurs="1">
                              <complexType>
                                 <sequence>
                                    <group ref="cel-core:partyBasicGroup"/>
                                 </sequence>
                              </complexType>
                           </element>
                           <element ref="dsig:Signature" minOccurs="1"/>
                        </sequence>
                     </sequence>
                  </complexType>
               </element>
            </sequence>
         </choice>
         <element name="Address" type="string" minOccurs="0"/>
      </sequence>
      <element name="EncryptedParty" type="cel-core:EncryptedPartyType"/>
   </choice>
   <attribute name="id" type="ID" use="required"/>
</complexType>
```

7.2.5.2 Example

The following contract exemplifies two parties, without signatures. The first party is an organization, represented by a signatory person who is further described by a virtual card. Note the use of RDF, used to give enriched information on the party. The second party is a person, whose details are given by means of the MPEG7:Person element.

```
<cel-core:Party id="RAI">
   <cel-core:Organization>
       <cel-core:Name>RAI Radiotelevisione Italiana</cel-core:Name>
       <dc:identifier>urn:VATIN:IT06382641006</dc:identifier>
       <dc:description>The Italian public broadcasting company</dc:description>
       <cel-core:Signatory>
             <cel-core:Name>Mario Rossi</cel-core:Name>
             <cel-core:Details xmlns:vCard="http://www .w3 .org/2001/vcard-rdf/3.0#"
xmlns:rdf="http://www .w3 .org/1999/02/22-rdf-syntax-ns#">
                <vCard:EMAIL rdf:parseType="Resource">
                   <rdf:value>thee@nowhere.com </rdf:value>
                   <rdf:type rdf:resource="http://www.w3.org/2001/vcard-rdf/3.0#inter-
net"/>
                </vCard:EMAIL>
             </cel-core:Details>
          </cel-core:Signatory>
    </cel-core:Organization>
    <cel-core:Address>Viale Mazzini 14, 00195 Roma, ITALY</cel-core:address>
</cel-core:Party>
<cel-core:Party id="JDOE">
    <cel-core:Person>
       <cel-core:Name>John Doe</cel-core:Name>
       <dc:identifier>urn:VATIN:123456</dc:identifier>
       <dc:description>Second party of the contract</dc:description>
       <cel-core:Details xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001">
          <mpeg7:Person>
             <mpeg7:Name>
                <mpeg7:GivenName>John</mpeg7:GivenName>
                <mpeg7:FamilyName>Doe</mpeg7:FamilyName>
                <mpeg7:Title>Dr</mpeg7:Title>
             </mpeg7:Name>
             <mpeg7:Affiliation>
                <mpeg7:Organization>John Doe Communications</mpeg7:Organization>
             </mpeg7:Affiliation>
             <mpeg7:Citizenship>UK</mpeg7:Citizenship>
             <mpeg7:ElectronicAddress>
                <mpeg7:Telephone>+12 345 6789100</mpeg7:Telephone>
                <mpeg7:Email>johndoe@secondparty.com</mpeg7:Email>
             </mpeg7:ElectronicAddress>
          </mpeg7:Person>
       </cel-core:Details>
    </cel-core:Person>
</cel-core:Party>
```

7.2.6 Body element

7.2.6.1 General

The cel-core:Body element contains the contract itself, i.e. the agreed clauses. It contains the following elements, as depicted in Figure 8:

- cel-core:TextualPart, optionally including a number of structured narrative clauses.
- cel-core:OperativePart, to be used, for example, in a MPEG-21 based system. It contains the deontic expressions of the operative part by means of XML representation as defined in 7.3.

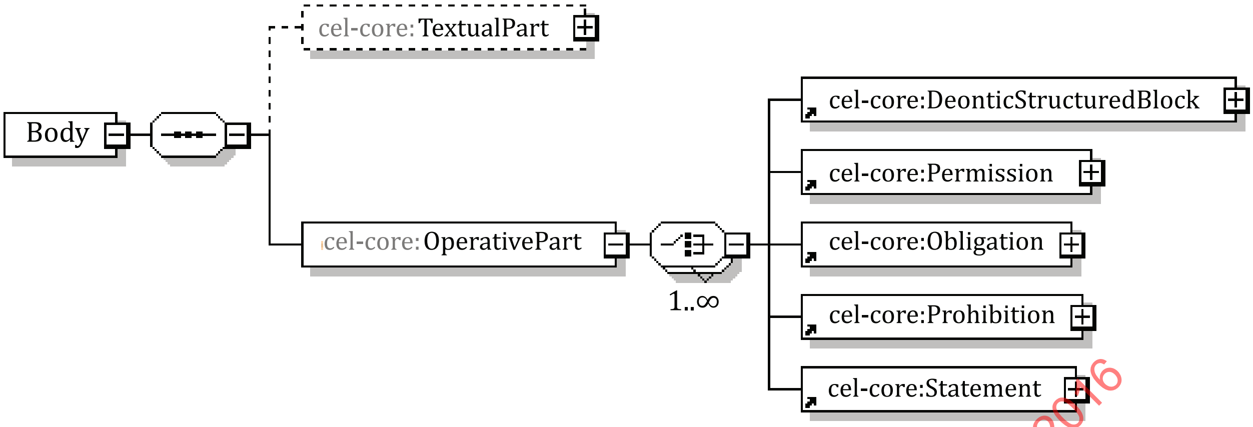

Figure 8 — Structure and placement of Body element

7.2.6.2 Text clauses

7.2.6.2.1 Overview

Text clauses are represented with elements of type `cel-core:TextClauseType`, which allows a structure of other nested `cel-core:TextClause` elements ending with an element `cel-core:TextParagraph`, containing the actual text. Text paragraphs can indistinctly appear in clear text or encrypted under the cel-`core:TextParagraph` or `cel-core:EncryptedTextParagraph` elements, respectively. The structure defined for the `cel-core:TextClause` element is depicted in Figure 9.

If the `cel-core:TextualPart` element contains any `cel-core:EncryptedTextParagraph` element, the `cel-core:Contract` element can contain an encrypted version of the whole narrative contract under the `cel-core:EncryptedTextVersion` element.

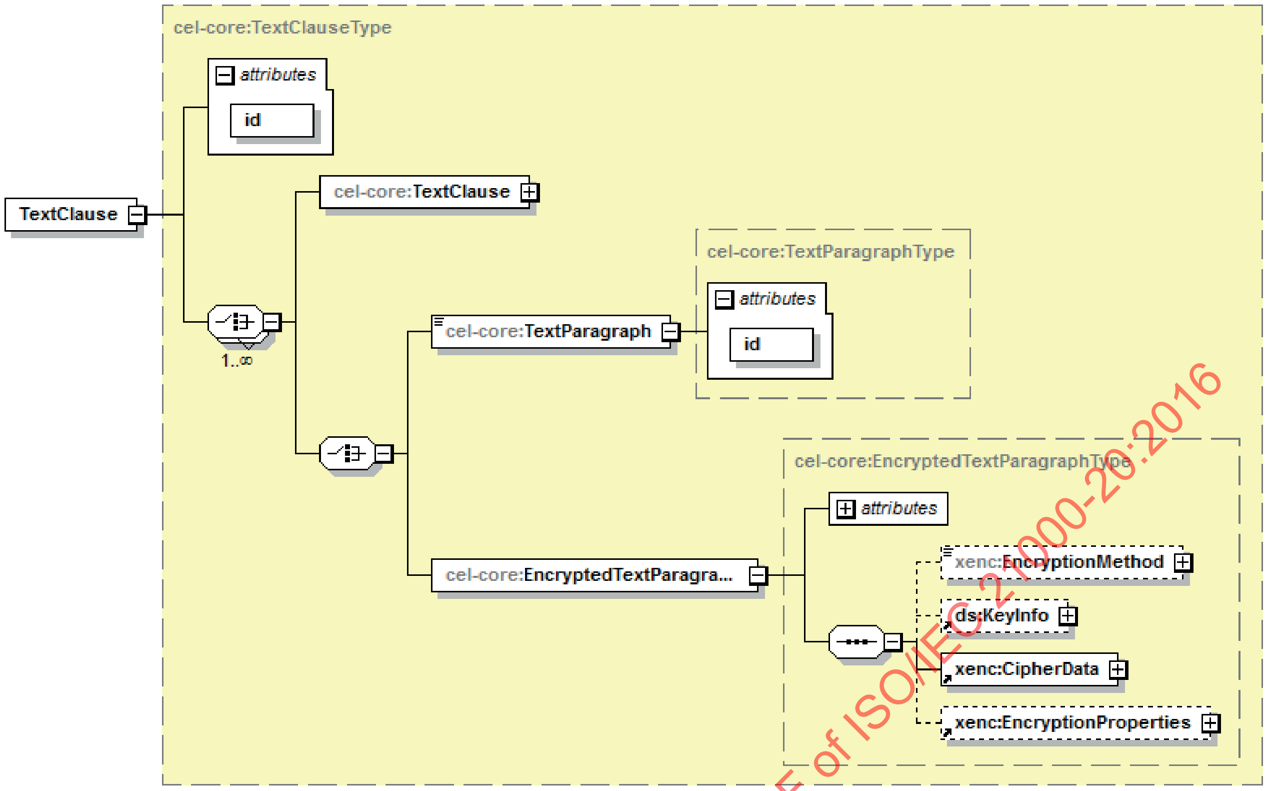

Figure 9 — TextClause element

The following box specifies the `cel-core:TextClauseType`, `cel-core:TextParagraphType` and the `cel-core:EncryptedTextParagraphType` types.

```
<complexType name="TextClauseType">
   <choice maxOccurs="unbounded">
      <element name="TextClause" type="cel-core:TextClauseType"/>
      <choice>
         <element name="TextParagraph" type="cel-core:TextParagraphType"/>
         <element name="EncryptedTextParagraph" type="cel-core:EncryptedTextPara-
graphType"/>
      </choice>
   </choice>
   <attribute name="id" type="ID" use="required"/>
</complexType>
<complexType name="TextParagraphType">
   <simpleContent>
      <extension base="string">
         <attribute name="id" type="ID" use="required"/>
      </extension>
   </simpleContent>
</complexType>
<complexType name="EncryptedTextParagraphType">
   <complexContent>
      <extension base="xenc:EncryptedDataType"/>
   </complexContent>
</complexType>
```

Both the `cel-core:TextClause` and the `cel-core:TextParagraph` elements have an attribute id of type `xsd:ID` which uniquely identifies them within the contract and allows to have a reference to them from the operative part section.

7.2.6.2.2 Text references

7.2.6.2.2.1 General

The reference to elements of the `cel-core:TextualPart` is achieved by means of the attribute idrefs of type IDREFS of the elements of type `cel-core:DeonticStructuredClauseType` or the element `cel-core:DeonticStructuredBlock`. The value of this attribute is the space-separated list of the IDs of the referenced elements. The XML validation will ensure that all the referenced elements are present in the document.

7.2.6.2.2.2 Example

The next example illustrates the usage of the CEL referencing mechanism, where the `cel-core:TextClause` and `cel-core:TextParagraph` referenced elements are those of the example in 7.2.6.2. The details of the elements `cel-core:Permission`, marked with the character '+', are not shown.

```
<cel-core:OperativePart>
 + <cel-core:Permission idrefs="TC003.3">
 + <cel-core:Permission idrefs="TP002R3 TP002L TP002F TP002ii">
 + <cel-core:Permission idrefs="TP002R2 TP002L TP002i">
 + <cel-core:Permission idrefs="TP002R1 TP002L TP002i">
</cel-core:OperativePart>
```

7.2.6.2.3 Encryption

Encrypted deontic structured blocks are represented by the `cel-core:EncryptedBlock` element of type `cel-core:EncryptedBlockType` which makes use of the XML Encryption.

```
<complexType name="EncryptedBlockType">
   <complexContent>
      <extension base="xenc:EncryptedDataType"/>
   </complexContent>
</complexType>
```

Encrypted operative clauses are represented by the `cel-core:EncryptedClause` element of type `cel-core:EncryptedClauseType` which makes use of the XML Encryption.

```
<complexType name="EncryptedClauseType">
   <complexContent>
      <extension base="xenc:EncryptedDataType"/>
   </complexContent>
</complexType>
```

7.2.6.2.4 Example

The next example represents in a structured way the contract text by using the cel-core:TextClause and the cel-core:TextParagraph elements, while the example in 7.2.6.2.1.2 shows an example of how these elements are referenced from the operative part.

```
<cel-core:TextClause id="TC002">
   <cel-core:TextParagraph id="TP002h">2.1 (Acquisition of 100% exploitation rights both
by free of charge and upon any kind of payment communication to the public and/or by mak-
ing the Program available to the public, both free of charge and upon any kind of pay-
ments, at the time and place chosen by the viewer, in Italy, Vatican City, Republic of
San Marino and

Principality of Monaco) Under this Agreement, Rai acquires
...</cel-core:TextParagraph>
   <cel-core:TextParagraph id="TP002i">i) by communication to the public through remote
diffusion/broadcast, whether
...</cel-core:TextParagraph>
<cel-core:TextParagraph id="TP002ii">ii) by making the Program available to the public
...</cel-core:TextParagraph>
   <cel-core:TextParagraph id="TP002F">The herein licensed rights shall also include the
exclusive right to place the Programs at the publics disposal with or without any payments

...</cel-core:TextParagraph>
   <cel-core:TextParagraph id="TP002L">The grant is made for n. 5 (five) years starting on
[...] , until [...], (the so-called License Period);
   </cel-core:TextParagraph>
   <cel-core:TextClause id="TC002R">
      <cel-core:TextParagraph id="TP002Rh">The following number of runs are hereby grant-
ed during the License Period:</cel-core:TextParagraph>
      <cel-core:TextParagraph id="TP002R1">- with reference to the communication to the
public by unencrypted and free of charge television
      ...</cel-core:TextParagraph>
      <cel-core:TextParagraph id="TP002R2">- with reference to the communication to the
public by encrypted and upon payment television diffusion/broadcast
      ...</cel-core:TextParagraph>

      <cel-core:TextParagraph id="TP002R3">- with reference to the right to make the Pro-
gram available to the public at the time and place chosen by the final viewer (point to
point communication)
      ...</cel-core:TextParagraph>
    </cel-core:TextClause>
</cel-core:TextClause>
<cel-core:TextClause id="TC003">
   <cel-core:TextClause id="TC003.3">
     <cel-core:TextParagraph id="TP003.3">3.3 (Right to use separate excerpts)
By the present Agreement, Rai is also granted the right to use and exploit in the Terri-
tory and during the License Period and during the License Period, even by the Internet or
similar networks (e.g. UMTS, etc.)
      ...</cel-core:TextParagraph>
   </cel-core:TextClause>
</cel-core:TextClause>
```

7.2.6.3 Operative part

The operative part is represented with a single container element, cel-core:OperativePart, which contains the deontic expressions of the operative part by means of the elements cel-core:DeonticStructuredBlock and/or any element of type cel-core:DeonticStructuredClauseType, i.e. cel-core:Permission, cel-core:Prohibition, cel-core:Obligation or cel-core:Statement. Both elements share the same mechanism for referencing elements in the cel-core:TextualPart and for allowing the inclusion of encrypted clauses. The structure of the cel-core:OperativePart element is shown in Figure 10. The element cel-core:DeonticStructuredBlock enables the grouping of contract clauses which are related in some way.

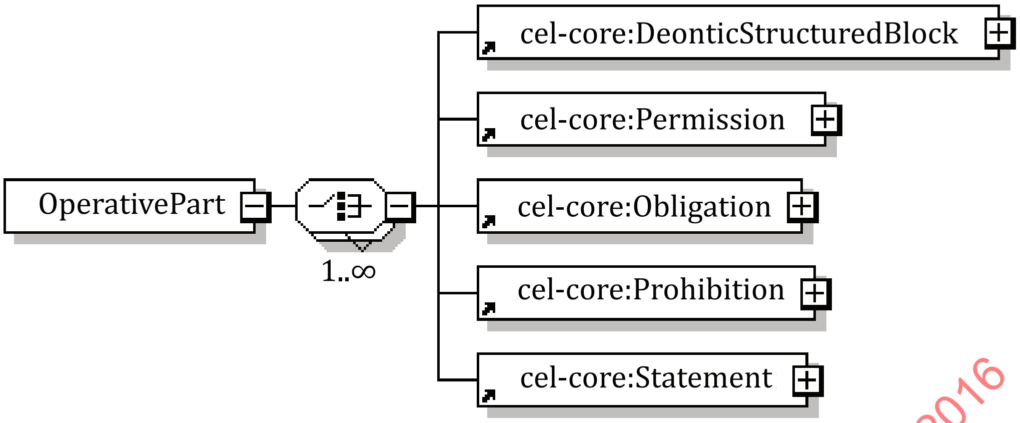

Figure 10 — OperativePart element

```
<element name="OperativePart">
   <complexType>
      <choice maxOccurs="unbounded">
         <element ref="cel-core:DeonticStructuredBlock"/>
         <element ref="cel-core:Permission"/>
         <element ref="cel-core:Obligation"/>
         <element ref="cel-core:Prohibition"/>
         <element ref="cel-core:Statement"/>
      </choice>
   </complexType>
</element>
```

```
<complexType name="DeonticStructuredClauseType">
   <choice>
      <sequence>
         <element ref="cel-core:Metadata" minOccurs="0"/>
         <element name="Context" type="anyType" minOccurs="0"/>
         <element name="PreCondition"
          type="cel-core:PreConditionType" minOccurs="0"
          maxOccurs="unbounded"/>
         <element ref="cel-core:Subject"/>
         <element name="Act" type="cel-core:ActType"/>
         <element name="Object" type="cel-core:ObjectType"
          minOccurs="0" maxOccurs="unbounded"/>
         <element name="ResultantObject" type="cel-core:ItemType"
          minOccurs="0" maxOccurs="unbounded"/>
         <element name="Constraint" type="cel-core:ConstraintType"
          minOccurs="0"/>
         <element name="PostCondition" type="cel-core:PostConditionType"
          minOccurs="0"/>
         <element name="Issuer" type="cel-core:IssuerType" minOccurs="0"/>
         </sequence>
         <element name="EncryptedClause" type="cel-core:EncryptedClauseType"/>
      </choice>
    <attribute name="number" type="string"/>
    <attribute name="id" type="ID" use="required"/>
    <attribute name="idrefs" type="IDREFS"/>
</complexType>
```

7.3 CEL Deontic Structure

7.3.1 General

MPEG-21 contracts contain agreements between parties, which are represented by deontic concepts of permission, prohibition, obligation and statement. To represent these agreements in XML language, blocks and clauses are used to represent operative parts of contracts in a machine-readable format. They are grouped in the `cel-core:OperativePart` element, described in 7.2.6.3, child element of the `cel-core:Body`. In the following subclauses, the elements comprising the CEL deontic structure are described in detail.

7.3.2 Deontic Structured Block

The `cel-core:DeonticStructuredBlock` is defined to group related deontic clauses as a nested structure. It is made up of `cel-core:DeonticStructuredBlock`, `cel-core:Permission`, `cel-core:Prohibition`, `cel-core:Obligation`, `cel-core:Statement` or `cel-core:EncryptedBlock` elements. Figure 11 shows its structure.

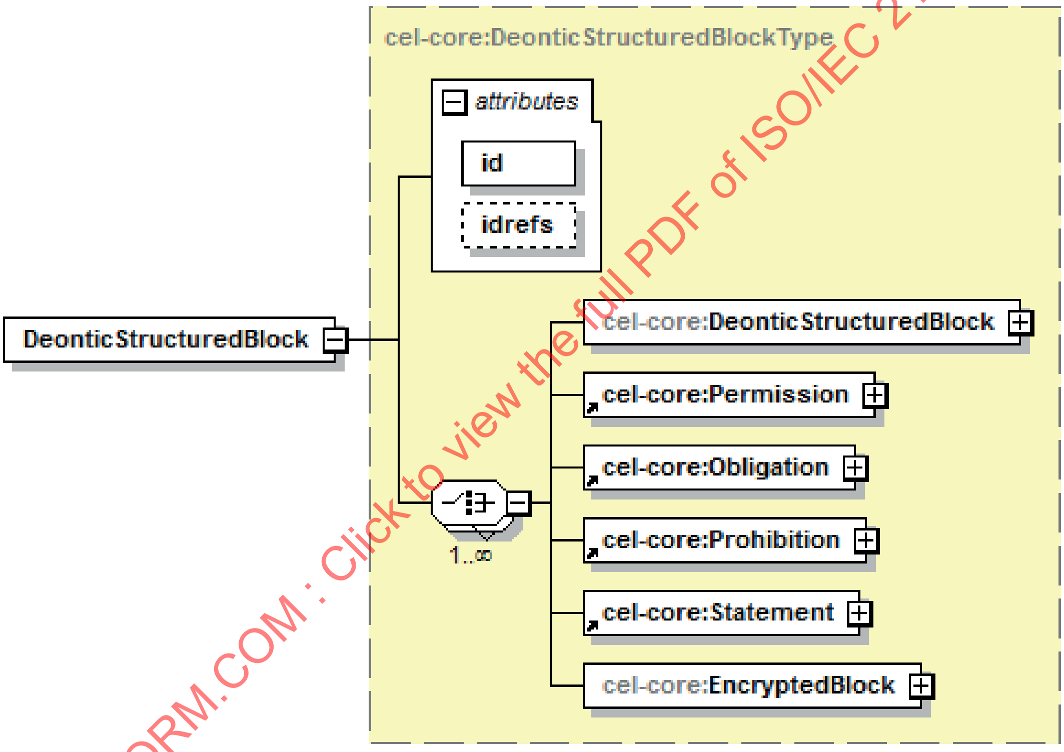

Figure 11 — DeonticStructuredBlock element

The `cel-core:DeonticStructuredBlock` element has the attributes:

— `id` — to uniquely identify the Block. This attribute is mandatory. It has type `xsd:ID`.

— `idrefs` — to optionally reference to textual parts of the contract as defined in 7.2.6.2.1.

The `cel-core:EncryptedBlock` element, if present, contains the encrypted version of a `cel-core:DeonticStructuredBlock` element. It makes use of the XML Encryption (see 7.2.6.2.3).

7.3.3 Statement

cel-core:Statement element defines the declaration of the matters of fact for the contract being defined. Although it is structured as a deontic clause, see 7.3.4, it has no deontic consequences for the actions by the parties.

7.3.4 Deontic Structured Clause Type

7.3.4.1 General

All the deontic clauses within the operative part of a contract, namely cel-core:Permission, cel-core:Obligation, cel-core:Prohibition or cel-core:Statement, are represented by the cel-core:DeonticStructuredClauseType, which consists of the following elements as depicted in Figure 12:

— cel-core:Metadata — XML deontic clause metadata;

— cel-core:Context — context for the XML deontic clause;

— cel-core:PreCondition — pre-conditions of the XML deontic clause;

— cel-core:Subject — entity to which the XML deontic clause applies;

— cel-core:Act — right to be applied;

— cel-core:Object — resources against which the right will apply;

— cel-core:ResultantObject — resource resultant of applying the cel-core:Act over the cel-core:Object;

— cel-core:Constraint — conditions which shall be fulfilled;

— cel-core:PostCondition — post-conditions of the deontic structured clause;

— cel-core:Issuer — party that issues the permission, prohibition, obligation or statement specified in the deontic clause.

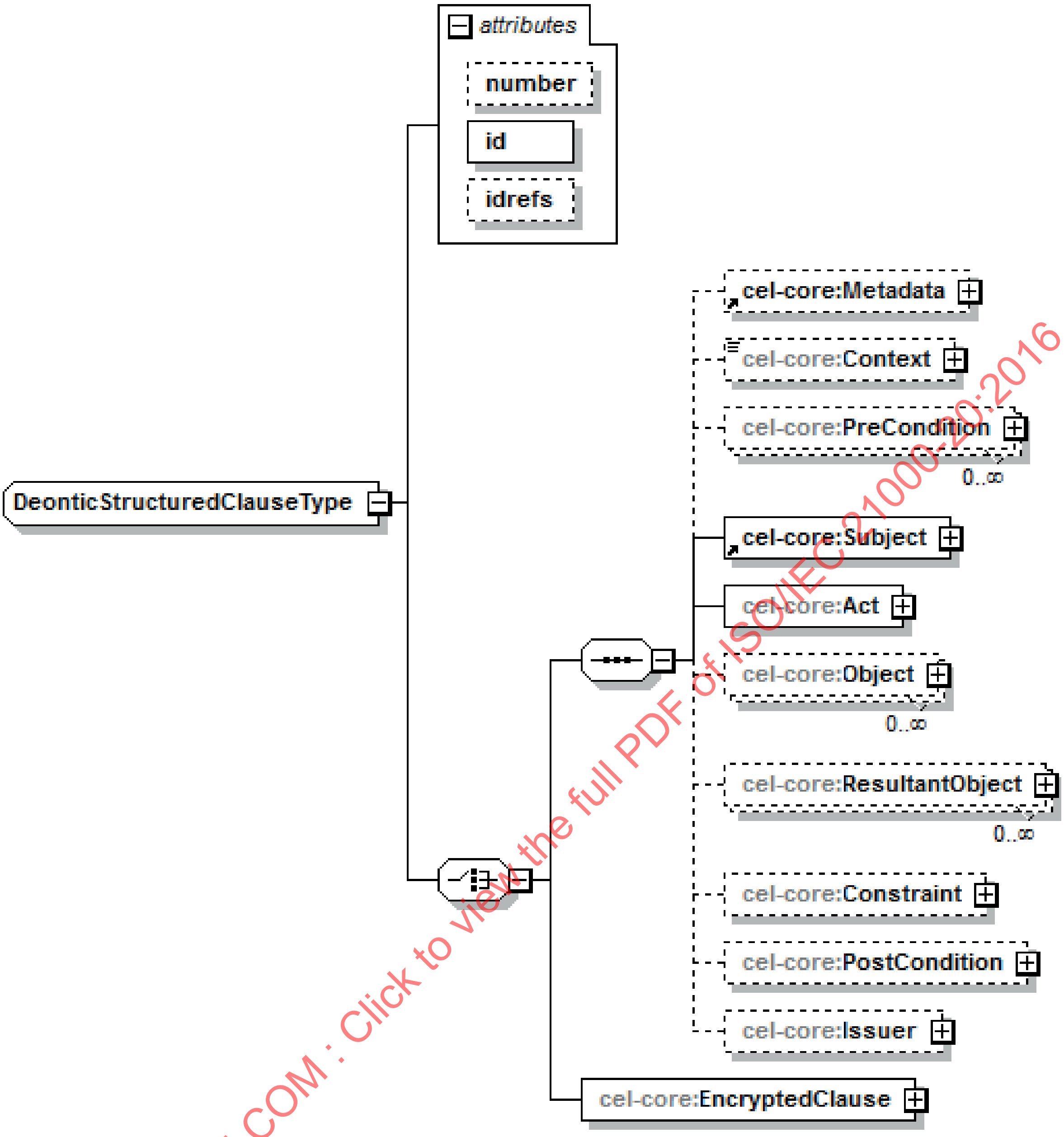

Figure 12 — DeonticStructuredClauseType type

The cel-core:DeonticStructuredClauseType type has the following attributes.

— number — number of the deontic structured clause. This attribute is optional. It has type string.

— id — to uniquely identify the clause. This attribute is mandatory. It has type xsd:ID.

— idrefs — to refer to the text clause(s) that represents. It has type IDREFS.

If the clause is encrypted, the cel-core:EncryptedClause element is present containing the encrypted version of a cel-core:DeonticStructuredClause element. It makes use of the XML Encryption.

The cel-core:DeonticStructuredClause element can be specialized in the following types of clauses: cel-core:Permission, cel-core:Obligation and cel-core:Prohibition, described in 7.3.4.2. There is also another clause type, cel-core:Statement, described in 7.3.3.

7.3.4.2 Permission, obligation and prohibition

This subclause describes the meaning of the specializations of the `cel-core:DeonticStructuredClause` element, which are as follows.

- `cel-core:Permission` element models the authorization from one rights owner to one or more users to realize one or more actions over one or more objects.
- `cel-core:Obligation` element defines that one of the parties has to execute an Action. The obligated action shall occur in the context of validity expressed by the constraints, once or continuously depending on the kind of action. It is also possible to require the obligated action to occur more than once by means of specifically defined constraints, such as `cel-core:Runs`.
- `cel-core:Prohibition` element defines that one of the parties cannot execute an action.

7.3.4.3 Metadata

Metadata can be added to a deontic structured clause under the `cel-core:Metadata` element, giving additional information about the clause itself.

The `cel-core:Metadata` element has been defined in 7.2.3, its structure is shown in Figure 13.

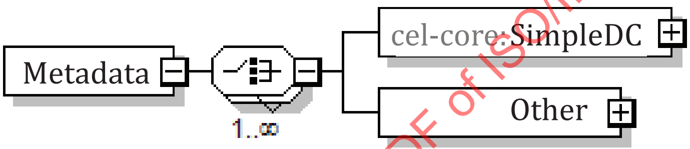

Figure 13 — Metadata element

7.3.4.4 Context

Contextual information of any type can be added to a deontic structured clause under the `cel-core:Context` element. This information is expressed as child element or attribute of `cel-core:Context` as depicted in Figure 14.

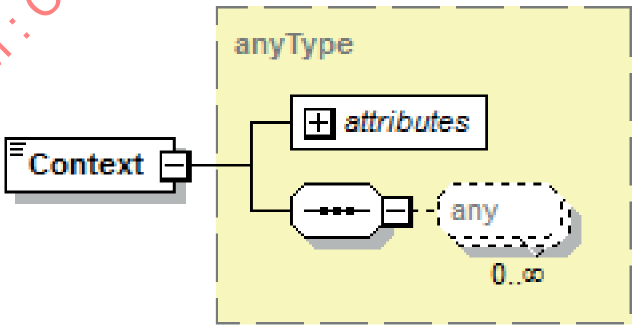

Figure 14 — Context element

7.3.4.5 PreCondition

The purpose of the `cel-core:PreCondition` element, together with the `cel-core:PostCondition` element, see 7.3.4.10, is to specify what an act accomplishes. A precondition indicates what should be true before the associated act is performed, whereas a postcondition indicates what will be true when performance of the act is finished.

The use of preconditions and postconditions allows to relate deontic clauses between them, indicating which depends on the other. It is even possible to define a chain of deontic clauses using pre/postcondition accordingly. Nevertheless, this is not the only way to do so. The `cel-core:Event` constraint, see 7.3.4.9.3 for details, also allows to relate a deontic clause to an action or an event. Both mechanisms express deontic clause interrelationship and the use of one mechanism or the other allows the construction of CEL contracts with equivalent properties. The decision on which to use just depends on the structure of the original contract.

Preconditions in a narrative contract clause are represented in a CEL contract by a deontic structured clause using the `cel-core:DeonticStructuredBlock` or `cel-core:DeonticStructuredClause` elements.

Once defined, pre-conditions can be stated in deontic clauses by means of the `cel-core:PreCondition` element, referring to an existing `cel-core:DeonticStructuredBlock` or `cel-core:DeonticStructuredClause` by means of the idref attribute, as depicted in Figure 15.

The `cel-core:PreCondition` also has the following attributes.

- `actionStatus` to optionally indicate the status of the action in the deontic expression to which the `cel-core:PreCondition` refers. If present, it shall take one of the two following values: ActionStarted or ActionDone. It has type `cel-core:ActionStatusType` (see Annex A).
- `withDelay` to optionally determine the elapsed time, following to the achievement of the `cel-core:Action` in the `cel-core:PreCondition` after which the permission, obligation or prohibition granted by the deontic expression to which the `cel-core:PreCondition` pertains has to be considered valid.
- `validity` to optionally determine the time of validity of the permission, obligation or prohibition granted by the deontic expression to which the `cel-core:PreCondition` pertains.

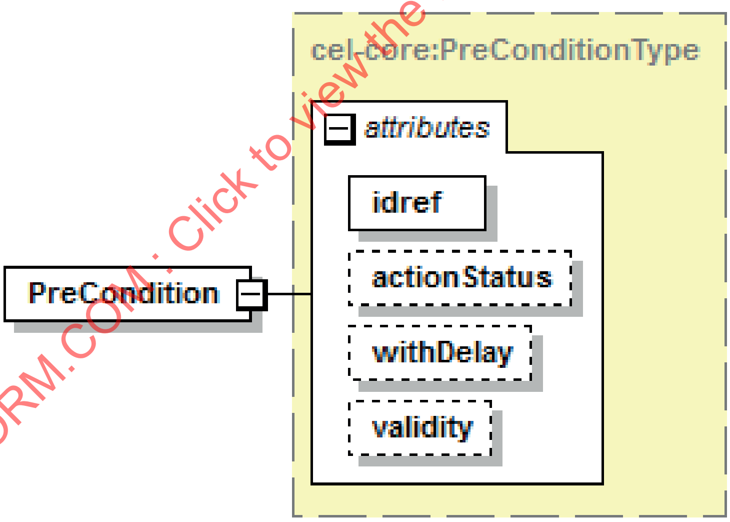

Figure 15 — PreCondition element

7.3.4.6 Subject

7.3.4.6.1 General

The `cel-core:Subject` element specifies the entity to which the deontic clause applies. It makes reference to one of the existing `cel-core:Party` elements of the contract by means of the `partyRef` attribute, as shown in Figure 16. In particular, it references to the party that may/should/should not/did execute the act, depending on if it is a permission, obligation, prohibition, or statement.

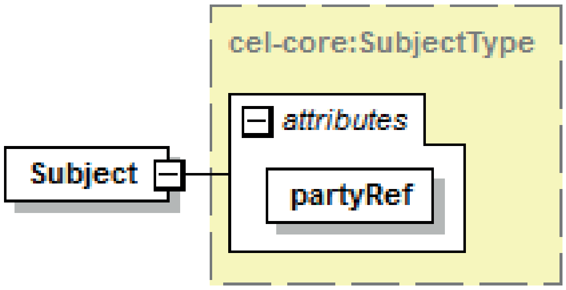

Figure 16 — Subject element

7.3.4.6.2 Example

```
<cel-core:Contract>
 ...
 <cel-core:Party id="user001"> ... </cel-core:Party>
 <cel-core:Party id="user002"> ... </cel-core:Party>
 <cel-core:Body>
  <cel-core:OperativePart>
   <cel-core:Permission id="p001">
    <cel-core:Subject partyRef="user002"/>
    <cel-core:Act> ... </cel-core:Act>
    <cel-core:Object> ... </cel-core:Object>
    ...
   </cel-core:Permission>
  </cel-core:OperativePart>
 </cel-core:Body>
</cel-core:Contract>
```

7.3.4.7 Act

7.3.4.7.1 General

The `cel-core:Act` element specifies the right that is permitted/obligated/prohibited to the `cel-core:Subject` if the constraints, pre-conditions and post-conditions (if present) are met.

The `cel-core:Act` has the following attribute:

- `id` — to uniquely identify the Act. This attribute is optional. It has type `xsd:ID`.

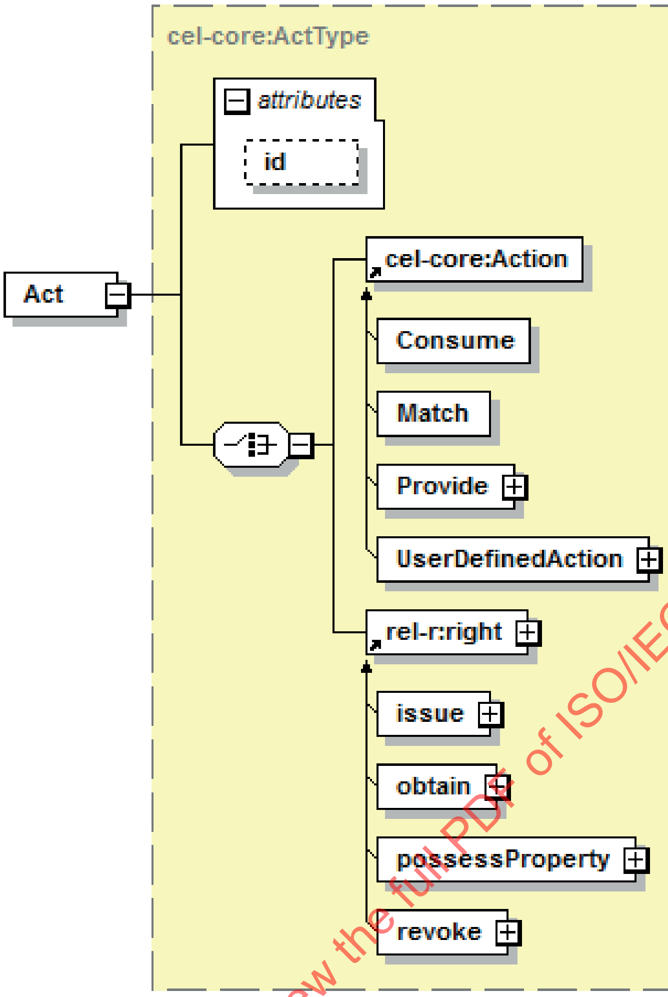

Figure 17 — Act element

The acts in a media contract include those defined in the CEL Extension for Exploitation of Intellectual Property Rights, see 7.4.2, under the `cel-core:Act` element replacing the `cel-core:Action` element or in ISO/IEC 21000-5 replacing the `rel-r:right` element. To this end, the `cel-core:ActType` has been defined as depicted in Figure 17.

7.3.4.7.2 Acts defined in core

Some of the acts required in contracts always apply, so they can be defined in the core model. Their semantic is defined in the following subclauses.

7.3.4.7.3 Provide

The `cel-core:Provide` element specifies the act of either offering a service, when the object of the act is a service, or that of transferring the content material to the parties referenced by the recipients attribute of type `IDREFS`. The parties shall be previously defined in the `cel-core:Party` element. If the content is provided in loan form, then the `isOnLoan` attribute shall be present with a true value.

The structure for the `cel-core:Provide` element is depicted in Figure 18.

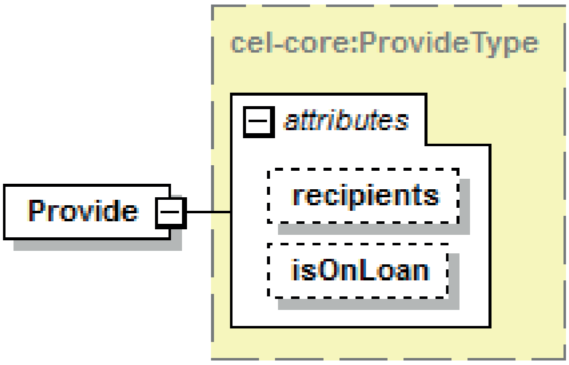

Figure 18 — Provide act

7.3.4.7.4 Consume

The `cel-core:Consume` element specifies the act of consuming a service or content.

7.3.4.7.5 Match

The `cel-core:Match` element models the action of executing a comparison between two (or more) objects or services and evaluating the degree of matching of the respective properties. The action may result in either a simple Boolean value or more complex structured information including ranking, confidence, or statistical data, depending on the application context. The input objects or services may play a different role in the action execution, e.g. one may be used as query or sample and another one as candidate under matching evaluation.

7.3.4.7.6 Example

The example below shows how an action has to be included inside a contract.

```
<cel-core:Contract>
 ...
  <cel-core:Body>
   <cel-core:OperativePart>
    <cel-core:Permission id="p001">
     <cel-core:Subject> ... </cel-core:Subject>
     <cel-core:Act>
      <cel-ipre:MakeAdaptation/>
     </cel-core:Act>
     <cel-core:Object> ... </cel-core:Object>
     ...
   </cel-core:Permission>
  </cel-core:OperativePart>
 </cel-core:Body>
</cel-core:Contract>
```

7.3.4.7.7 Extending contract acts: UserDefinedAction element

This subclause describes the mechanism for including new user defined acts in MPEG-21 Contracts. The `cel-core:UserDefinedAction` element allows the creation of new user defined actions.

The `cel-core:UserDefinedAction` element consists of the following elements, as depicted in Figure 19:

— `cel-core:Name` indicates the name of the new Action;

— `cel-core:StandardReference` to reference to a standard part;

— `cel-core:Definition` to define the semantics of the action inside the XML schema.

It is worth noting that only one of the elements `cel-core:StandardReference` or `cel-core:Definition` shall be present.

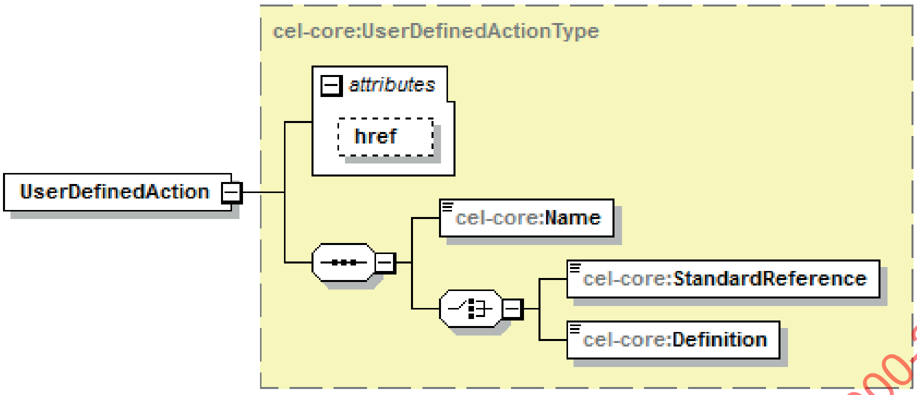

Figure 19 — UserDefinedAction element

The `cel-core:UserDefinedAction` has the following attribute:

— `href` to reference the standard part or public website where the element definition can be found.

The following is an example of the definition of a new User Defined Action that represents the "DELETE" action in the e-Health domain.

```
<cel-core:Contract>
 ...
 <cel-core:Body>
  <cel-core:OperativePart>
   <cel-core:Permission id="p001">
    <cel-core:Subject> ... </cel-core:Subject>
    <cel-core:UserDefinedAction
      href="http://www .hl7 .org/implement/standards/product_brief.cfm ?
      product_id=72">
      <cel-core:Name>DELETE</cel-core:Name>
      <cel-core:StandardReference>
      HL7:security:group
      </cel-core:StandardReference>
    </cel-core:UserDefinedAction>
    <cel-core:Object> ... </cel-core:Object>
    ...
   </cel-core:Permission>
  </cel-core:OperativePart>
 </cel-core:Body>
</cel-core:Contract>
```

7.3.4.8 Object

7.3.4.8.1 General

The `cel-core:Object` element defines the resource against which the deontic expression will apply.

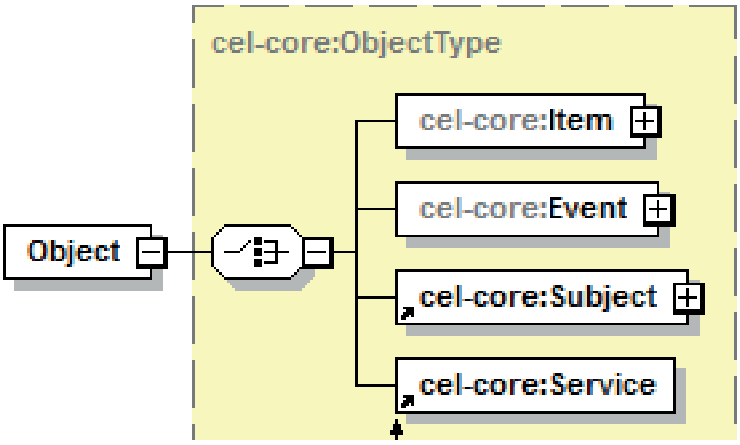

Figure 20 — Object element

Objects are defined in deontic clauses under the `cel-core:Object` element, as shown in Figure 20. The following types of objects are supported.

- A digital resource identified by the `dii:Identifier` element and/or by the `dii:RelatedIdentifier` element of the `cel-core:Item` element. It can be optionally named by means of the name attribute.
- A real life event, represented by element `cel-core:Event`, subject to intellectual property. Examples include sport events, concerts, and musical contexts. Events occur at a given time and thus their status can vary in time with respect to be started, in progress, suspended or finished. Event is here identified by at least one `dii:RelatedIdentifier`.
- One of the parties of the contract referenced by the `partyRef` attribute of the `cel-core:Subject` element.
- An MPEG service, like authentication, identification, encryption, processing, search or others (see 7.4.3.16) as defined in the `cel-core:Service` element.

7.3.4.8.2 Example

```
<cel-core:Contract>
 ...
 <cel-core:Body>
  <cel-core:OperativePart>
    <cel-core:Permission id="p001">
     <cel-core:Subject> ... </cel-core:Subject>
     <cel-core:Act> ... </cel-core:Act>
     <cel-core:Object>
     <cel-core:Item name="Title1">
      <dii:Identifier>it.rai:idteca:F00009</dii:Identifier>
     </cel-core:Item>
    </cel-core:Object>
    ...
   </cel-core:Permission>
  </cel-core:OperativePart>
 </cel-core:Body>
</cel-core:Contract>
```

7.3.4.8.3 Services

The services required for modeling rights in audiovisual contracts are depicted in Figure 21. The semantic for services is provided next. The services definitions are as follows:

- `cel-core:Authenticate` confirms the identity of a Content item in a value chain;

- `cel-core:Deliver` transfers Content between Users of a value chain;
- `cel-core:Describe` associates metadata to Content in a value chain;
- `cel-core:Identify` assigns Identifiers to Content in a value chain;
- `cel-core:InteractWith` triggers an action within a value chain through Content presented;
- `cel-core:Package` makes Content ready for delivery;
- `cel-core:Present` experiences Content;
- `cel-core:Post` lets other user access their Content;
- `cel-core:Process` performs operations on Content in a value chain;
- `cel-core:Store` saves Content for later use;
- `cel-core:Verify` checks the integrity of Content.

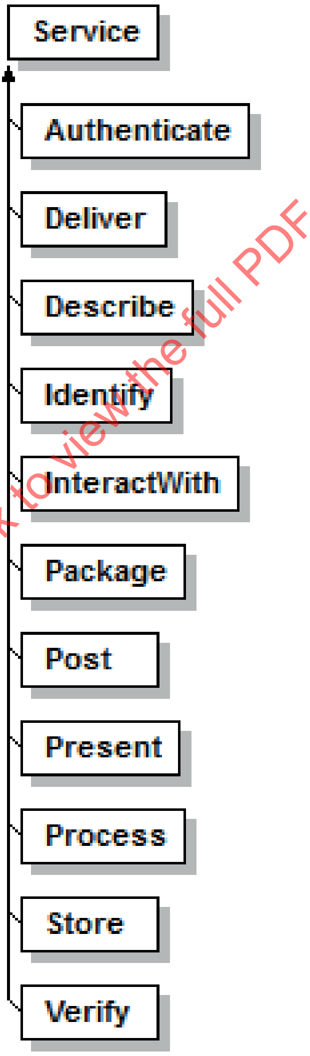

Figure 21 — CEL services

7.3.4.9 ResultantObject

7.3.4.9.1 General

The `cel-core:ResultantObject` represents the new digital resource which will result from the act of applying the right specified in the `cel-core:Act` element over the digital resource specified in the `cel-core:Object` element of a specific deontic clause, to which it pertains. The resultant resource is identified by the `dii:Identifier` and can be optionally named by means of the name attribute, as shown in Figure 22.

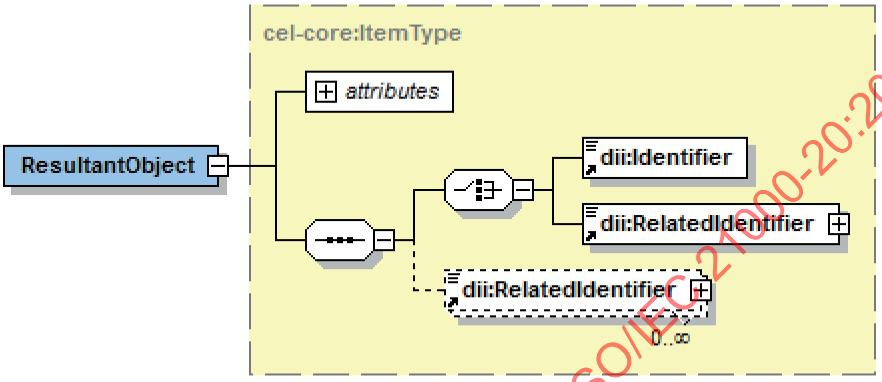

Figure 22 — ResultantObject element

7.3.4.9.2 Example

```
<cel-core:Contract>
  ...
  <cel-core:Body>
   <cel-core:OperativePart>
    <cel-core:Permission id="p001">
     <cel-core:Subject> ... </cel-core:Subject>
     <cel-core:Act> ... </cel-core:Act>
     <cel-core:Object> ... </cel-core:Object>
     <cel-core:ResultantObject name="NewAV">
      <dii:Identifier>it.rai:idteca:F00001#id=excerpt</dii:Identifier>
     </cel-core:ResultantObject>
     ...
   </cel-core:Permission>
  </cel-core:OperativePart>
 </cel-core:Body>
</cel-core:Contract>
```

7.3.4.10 Constraint

7.3.4.10.1 General

Conditions, restrictions and constraints can be specified under the `cel-core:Constraint` element, as depicted in Figure 23.

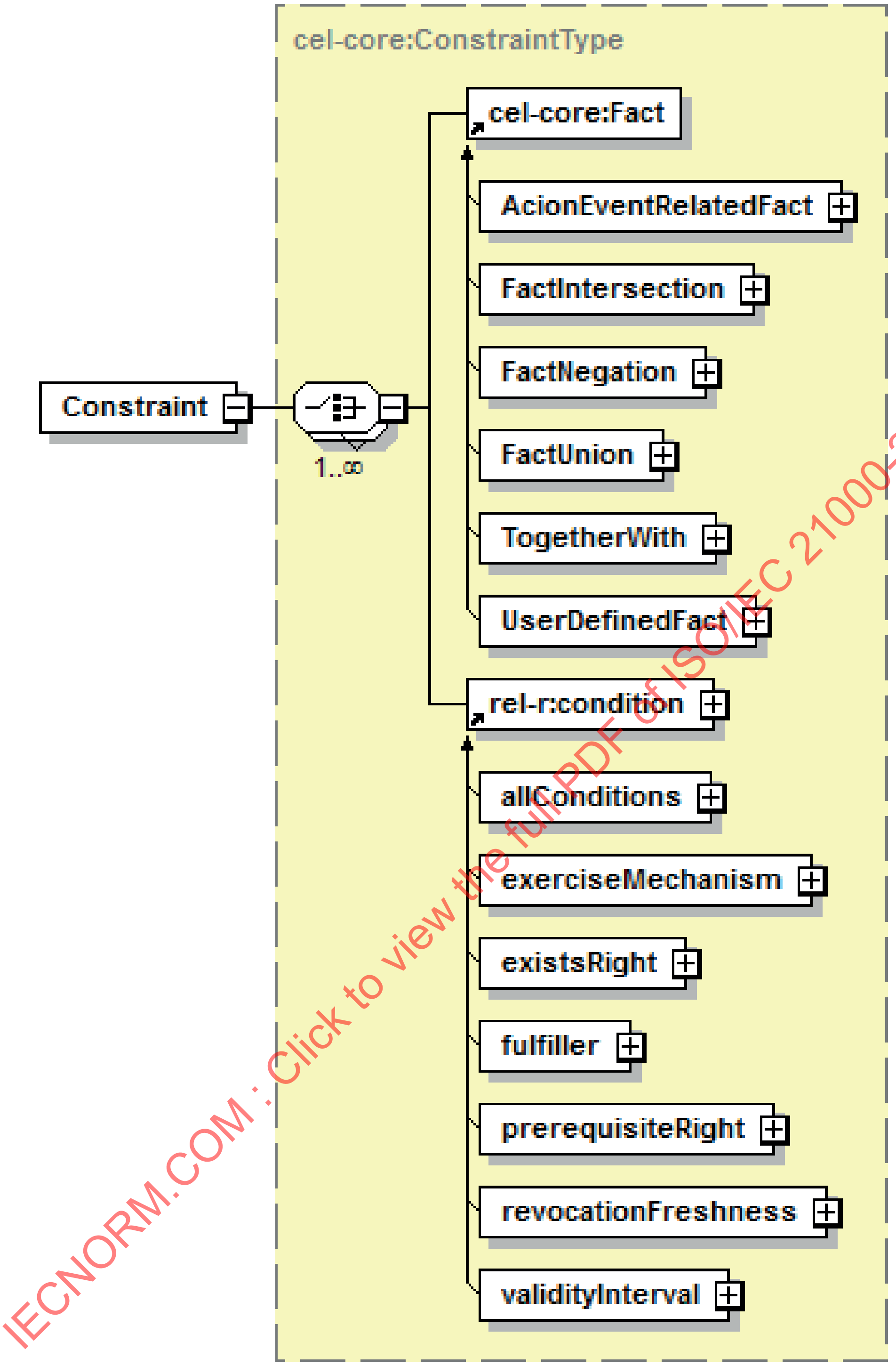

Figure 23 — Constraint element

Simple conditions, restrictions and constraints of a deontic clause include those defined in the CEL Extension for Exploitation of Intellectual Property Rights, see 7.4.3, under the `cel-core:Constraint`, `cel-core:FactIntersection`, `cel-core:FactNegation`, `cel-core:FactUnion`, `cel-core:TogetherWith`, `cel-core:ActionEventFact` or `cel-core:UserDefinedFact` elements replacing the `cel-core:Fact` element or in ISO/IEC 21000-5 replacing the `rel-r:condition` element. In fact, `cel-core:Constraint` works as a fact intersection as all conditions defined shall be satisfied.

Complex conditions are represented by means of the cel-core:FactIntersection, cel-core:FactNegation, cel-core:FactUnion and cel-core:TogetherWith elements.

The cel-core:FactIntersection element is used to represent the intersection of simple and/or complex conditions. Its evaluation will be true if the evaluation of all the facts which form it is true.

The cel-core:FactNegation element is used to represent the negation of simple or complex conditions. Its evaluation will be true if the evaluation of the only fact which forms it is false.

The cel-core:FactUnion element is used to represent the union of simple and/or complex conditions. Its evaluation will be true if the evaluation of at least one of the facts which form it is true.

The cel-core:TogetherWith element is to express conditions on which objects can be involved in the action, other than the one indicated in the deontic expression itself.

The cel-core:ActionEventFact element is to express conditions depending on the status of a related action or event.

The cel-core:UserDefinedFact element allows the creation of new simple user defined constraints, as described in 7.3.4.9.1.

7.3.4.10.2 Extending contract constraints: UserDefinedFact element

This subclause describes the mechanism for including new user defined constraints in MPEG-21 Contracts. The cel-core:UserDefinedFact element allows the creation of new user defined constraints.

The cel-core:UserDefinedFact element consists of the following elements, as depicted in Figure 24:

— cel-core:Name indicates the name of the new Fact;

— cel-core:StandardReference to reference to a standard part;

— cel-core:Definition to define the semantics of the fact inside the XML schema.

It is worth noting that only one of the elements cel-core:StandardReference or cel-core:Definition should be present.

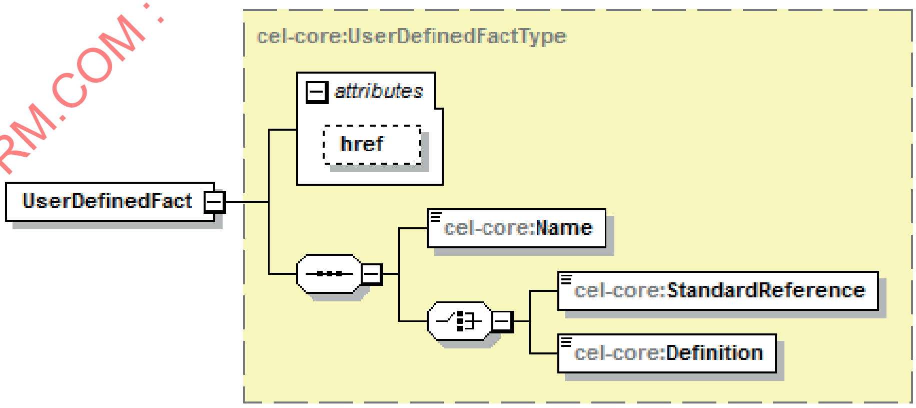

Figure 24 — UserDefinedFact element

The cel-core:UserDefinedFact has the attribute href to reference a standard part or public website where the element definition can be found.

7.3.4.10.3 User Defined Fact example

The following grey box shows an example of the definition of a user defined fact.

```
<cel-core:Contract>
 ...
 <cel-core:Body>
  <cel-core:OperativePart>
   <cel-core:Permission id="p001">
    <cel-core:Subject> ... </cel-core:Subject>
    <cel-core:Act> ... </cel-core:Act>
    <cel-core:Object> ... </cel-core:Object>
    <cel-core:UserDefinedFact>
      <cel-core:Name>seniorAuthorizationGiven</cel-core:Name>
      <cel-core:Definition>
         The authorization of a senior employee is required to perform the Action.
      </cel-core:Definition>
    </cel-core:UserDefinedFact>
   </cel-core:Permission>
  </cel-core:OperativePart>
 </cel-core:Body>
</cel-core:Contract>
```

7.3.4.10.4 ActionEventFact

The cel-core:ActionEventFact element is to express conditions depending on the status of a related action or event (see Figure 25).

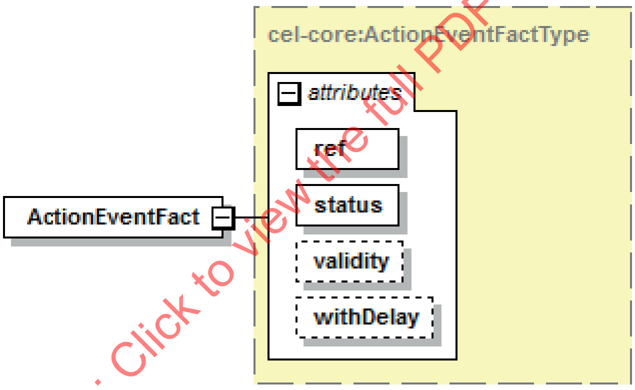

Figure 25 — ActionEventFact element

The cel-core:ActionEventFact has the following attributes:

— ref, required, shall contain a valid reference to either the related Act or the related Event, if possible as xsd:ID, otherwise as xsd:anyURI;

— status, required, indicates whether the condition relates to start or completion time of the action or event;

— validity, optional, indicates, if constrained, for how long the condition provides validity to the deontic expression;

— withDelay, optional, indicates the elapsed time following the start or completion of the related action or event after which the condition provides validity to the deontic expression.

7.3.4.10.5 Simple constraint example

The following grey box shows an example of a temporal Interval constraint.

```
<cel-core:Contract>
 ...
 <cel-core:Body>
  <cel-core:OperativePart>
   <cel-core:Permission id=”p001”>
    <cel-core:Subject> ... </cel-core:Subject>
    <cel-core:Act> ... </cel-core:Act>
    <cel-core:Object> ... </cel-core:Object>
    <cel-core:Constraint>
        <cel-ipre:TemporalContext
           afterDate=”2009-11-19T00:00:00”
           beforeDate=”2012-11-18T23:59:59”/>
    </cel-core:Constraint>
   </cel-core:Permission>
  </cel-core:OperativePart>
 </cel-core:Body>
</cel-core:Contract>
```

7.3.4.10.6 Complex constraint example

The following grey box shows an example of a complex constraint, which consists of the union of three constraints. The first, an intersection of the constraints Runs and DeliveryModality; the second, an intersection of the constraints DeliveryModality and AccessPolicy with value Pay; and the third, the DeliveryModality constraint. Figure 26 shows the relationship between the constraints graphically.

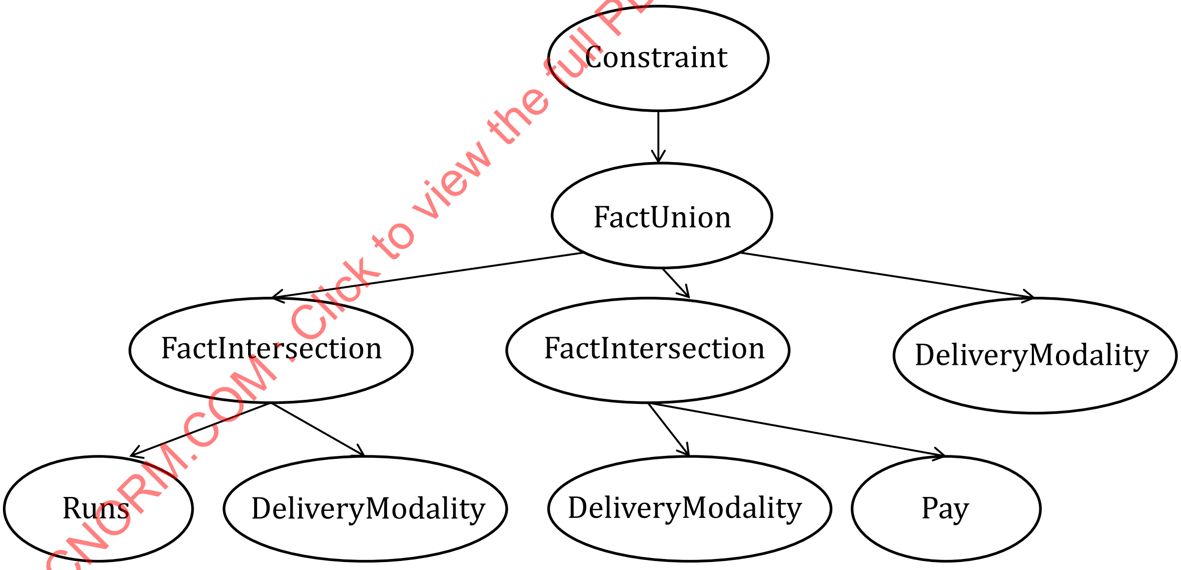

Figure 26 — Complex constraint graphical representation

```
<cel:Contract>
 ...
 <cel-core:Body>
  <cel-core:OperativePart>
   <cel-core:Permission id="p001">
    <cel-core:Subject> ... </cel-core:Subject>
    <cel-core:Act> ... </cel-core:Act>
    <cel-core:Object> ... </cel-core:Object>
    <cel-core:Constraint>
     <cel-core:FactUnion>
      <cel-core:FactIntersection>
       <cel-ipre:Runs numberOfRuns="10"/>
       <cel-ipre:DeliveryModality href="urn:mpeg:mpeg21:cel:ipre:2015/DeliveryModality#-
Linear"/>
      </cel-core:FactIntersection>
      <cel-core:FactIntersection>
       <cel-ipre:DeliveryModality href="urn:mpeg:mpeg21:cel:ipre:2015/DeliveryModality#-
Linear"/>
       <cel-ipre:AccessPolicy href="urn:mpeg:mpeg21:cel:ipre:2015/AccessPolicy#Pay"/>
      </cel-core:FactIntersection>
      <cel-ipre:DeliveryModality href="urn:mpeg:mpeg21:cel:ipre:2015/DeliveryModali-
ty#NonLinear"/>
     </cel-core:FactUnion>
    </cel-core:Constraint>
   </cel-core:Permission>
  </cel-core:OperativePart>
 </cel-core:Body>
</cel-core:Contract>
```

7.3.4.11 PostCondition

The purpose of the cel-core:PostCondition element, together with the cel-core:PreCondition element, see 7.3.4.4, is to specify what an act accomplishes. A post-condition indicates what will be true when performance of the act is finished, whereas a pre-condition indicates what should be true before the associated act is performed.

Post-conditions in a narrative contract clause are represented in a CEL contract by a deontic structured clause using the cel-core:DeonticStructuredBlock or cel-core:DeonticStructuredClause elements.

Once defined, post-conditions can be stated in deontic clauses by means of the cel-core:PostCondition element, referring to an existing cel-core:DeonticStructuredBlock or cel-core:DeonticStructuredClause by means of the idref attribute, as depicted in Figure 27.

The cel-core:PostCondition also has the following attributes.

- actionStatus to optionally indicate the status of the action in the deontic expression to which the cel-core:PostCondition refers. If present, it shall take one of the two following values: ActionStarted or ActionDone. It has type cel-core:ActionStatusType.

- withDelay to optionally determine the elapsed time, following to the achievement of the cel-core:Action in the cel-core:PostCondition after which the permission, obligation or prohibition granted by the deontic expression to which the cel-core:PosCondition pertains has to be considered valid.

- validity to optionally determine the time of validity of the permission, obligation or prohibition granted by the deontic expression to which the cel-core:PostCondition pertains.

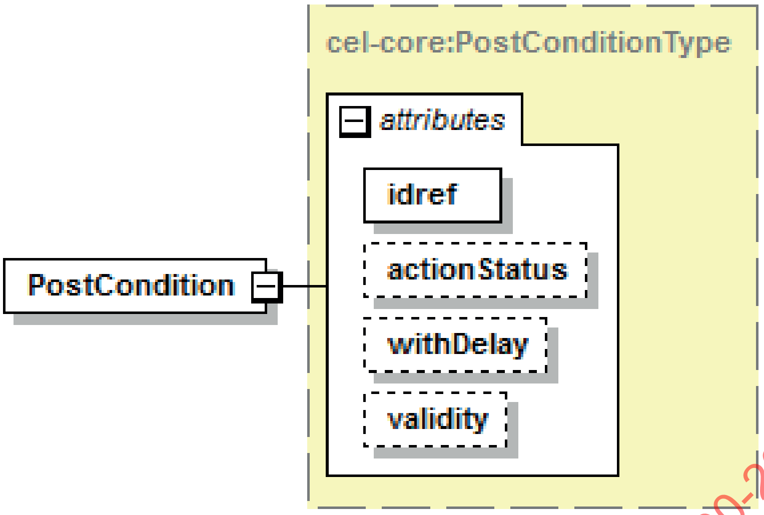

Figure 27 — PostCondition element

7.3.4.12 Issuer

The `cel-core:Issuer` element specifies the party in a CEL contract that issues the permission, prohibition, obligation or statement specified in the deontic clause to which it pertains.

It makes reference to one of the existing `cel-core:Party` elements of the contract by means of the `cel-core:partyRef` attribute, as shown in Figure 28.

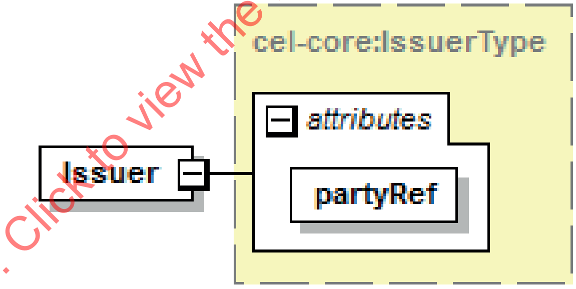

Figure 28 — Issuer element

7.4 CEL Extension for Exploitation of Intellectual Property Rights

7.4.1 General

The XML CEL Extension for Exploitation of Intellectual Property Rights is defined for representing the most common acts and constraints in the media field, to be later used in digital media contracts. The CEL IPRE extension namespace is `urn:mpeg:mpeg21:cel:ipre:2015`.

The reference data for such constraints are given in A.2 in the form of an MPEG-7 set of classification schemes, which list the reference terms in a structured hierarchical way and provide a URI-based reference mechanism.

The semantics of constraints, including the meaning implied by their position in the reference data structure, is given in the following subclauses.

7.4.2 Acts

7.4.2.1 General

The acts required for modeling rights in audiovisual contracts are defined according to a hierarchy which is depicted in Figure 29, and which is defined in A.1.2 by means of the cascade of `substitutionGroup` given below.

```
<element name="CommunicationToThePublic" type="cel-core:Action" substitutionGroup="-
cel-ipre:PublicCommunication"/>
<element name="CreativeTransform" type="cel-core:Action" substitutionGroup="cel-ip-
re:Transform"/>
<element name="Distribute" type="cel-core:Action" substitutionGroup="cel-ipre:ExploitI-
PRights"/>
<element name="Duplicate" type="cel-core:Action" substitutionGroup="cel-ipre:ExploitI-
PRights"/>
<element name="ExploitIPRights" type="cel-core:Action" substitutionGroup="cel-core:Ac-
tion"/>
<element name="Fixate" type="cel-core:Action" substitutionGroup="cel-ipre:ExploitI-
PRights"/>
<element name="MakeAdaptation" type="cel-core:Action" substitutionGroup="cel-ipre:Trans-
form"/>
<element name="MakeCopy" type="cel-core:Action" substitutionGroup="cel-ipre:Duplicate"/>
<element name="MakeCutAndEdit" type="cel-core:Action" substitutionGroup="cel-ipre:Trans-
form"/>
<element name="MakeExcerpt" type="cel-core:Action" substitutionGroup="cel-ipre:Trans-
form"/>
<element name="MakeRadioProduct" type="cel-core:Action" substitutionGroup="cel-ip-
re:Transform"/>

<element name="MakeInstance" type="cel-core:Action" substitutionGroup="cel-ipre:Fixate"/>
<element name="Novelization" type="cel-core:Action" substitutionGroup="cel-ipre:Creative-
Transform"/>
<element name="Prequel" type="cel-core:Action" substitutionGroup="cel-ipre:CreativeTrans-
form"/>
<element name="PublicCommunication" type="cel-core:Action" substitutionGroup="cel-ipre:-
ExploitIPRights"/>
<element name="PublicPerformance" type="cel-core:Action" substitutionGroup="cel-ipre:Pub-
licCommunication"/>
<element name="Remake" type="cel-core:Action" substitutionGroup="cel-ipre:CreativeTrans-
form"/>
<element name="Remix" type="cel-core:Action" substitutionGroup="cel-ipre:Transform"/>
<element name="Sequel" type="cel-core:Action" substitutionGroup="cel-ipre:CreativeTrans-
form"/>
<element name="Spinoff" type="cel-core:Action" substitutionGroup="cel-ipre:CreativeTrans-
form"/>
<element name="Translate" type="cel-core:Action" substitutionGroup="cel-ipre:Transform"/>
<element name="Transform" type="cel-core:Action" substitutionGroup="cel-ipre:ExploitI-
PRights"/>
```

When a deontic clause uses an act of the hierarchy, the deontic associated to all child acts of the given act is automatically applied.

EXAMPLE If deontic clause permits the act ‘Transform’, it is implied that also ‘MakeCutEdit’, ‘MakeExcerpt’, and ‘Remix’ are permitted.

This mechanism allows the parties to define deontic clauses with the desired degree of generality/specificity.

As ‘ExploitIPRights’ is the root term of the hierarchy, a permission to act ‘ExploitIPRights’ is equivalent to have all economical rights, with the limitations specified by the associated constraints.

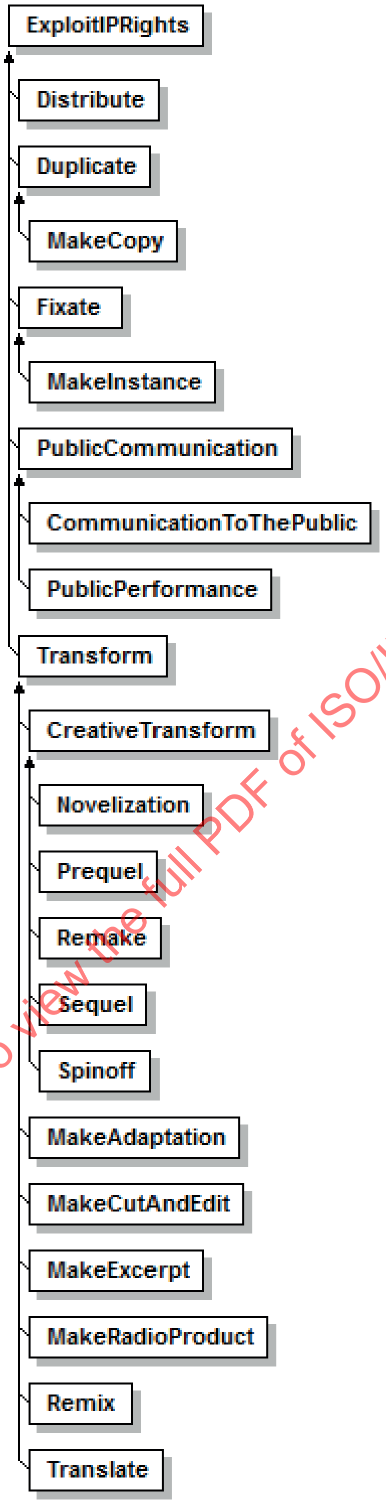

Figure 29 — Media Contracts CEL extension acts

The hierarchy of these actions and their semantic is described below, following a similar structure to the one defined in ISO/IEC 21000-21:2016, 8.3.2.

7.4.2.2 ExploitIPRights and related acts

The act cel-ipre:ExploitIPRights represents the action of exploiting the intellectual property rights. This act is the parent action of the following acts:

— cel-ipre:Distribute, for modeling the distribution of audiovisual material, which involves sell, rent and lend;

— cel-ipre:Duplicate, to make copies of the work; this is equivalent to cel-ipre:MakeCopy;

— cel-ipre:Fixate, which represents the action by which a work or a performance is materialy recorded; this is equivalent to cel-ipre:MakeInstance;

— cel-ipre:PublicCommunication, for representing the action of publicly displaying/performing, e.g. live performance, radio, television, Internet;

— cel-ipre:Transform, for modeling all forms of modifications.

7.4.2.3 PublicCommunication and related acts

The act cel-ipre:PublicCommunication is the parent action of the following acts:

— cel-ipre:CommunicationToThePublic, which models a public communication in which the public is not present in the place where the communication is originated;

— cel-ipre:PublicPerformance, which models a public communication in which the audiovisual material is showed or played in public.

7.4.2.4 Transform and related acts

The act cel-ipre:Transform, which models all forms of modifications, is the parent action of the following acts:

— cel-ipre:CreativeTransform, which models any transformation of an audiovisual material into a new one, containing new creative editorial elements;

— cel-ipre:MakeAdaptation, which represents the action of making an adaptation of an audiovisual material;

— cel-ipre:MakeCutAndEdit, which models cut, translation, editing, and what else alter the audiovisual content for customary exploitation purposes;

— cel-ipre:MakeExcerpt, which models the use and exploitation of excerpts separate from the source audiovisual content;

— cel-ipre:MakeRadioProduct, which models the transformation into an audiovisual material mainly made up of sound component;

— cel-ipre:Remix, which models the recombination of audio tracks or channels from a recording to produce a new or modified audio recording;

— cel-ipre:Translate, which models the modification of the audiovisual material language.

7.4.2.5 Creative Transform and related acts

The act cel-ipre:CreativeTransform, which models any transformation of an audiovisual material into a new one, is the parent action of the following acts:

— cel-ipre:Novelization, which models the transformation of the original audiovisual material into a new literary work;

- `cel-ipre:Prequel`, which models the transformation into a new audiovisual material, having same or related characters and/or same or related context elements than the original one, for which the fiction story is consistently conceived to happen before the original story;
- `cel-ipre:Remake`, which models the creation of a new audiovisual material with the same theme or concept or plot than the original one;
- `cel-ipre:Sequel`, which models the transformation into a new audiovisual material, having same or related characters and/or same or related context elements than the original one, for which the fiction story is consistently conceived to be the continuation of the original story;
- `cel-ipre:Spinoff`, which models the transformation into a new audiovisual material, having some common elements with the original one.

7.4.3 Constraints

7.4.3.1 General

The constraints required for modeling conditions in audiovisual contracts are depicted in Figure 30. The semantic for constraints is provided in next subclauses.

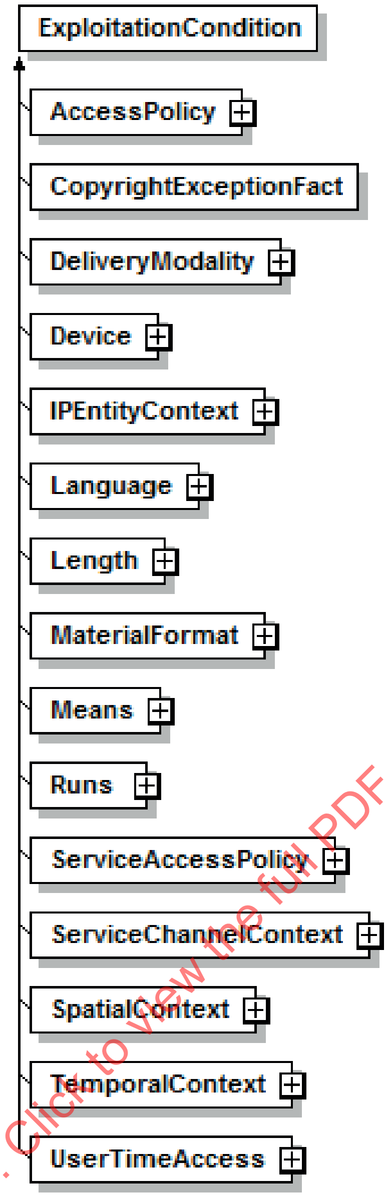

Figure 30 — Media Contracts CEL extension constraints

7.4.3.2 Constraints defined as fact hierarchy

Some constraints are defined in terms of a fact hierarchy, following a similar structure to the one defined in ISO/IEC 21000-21:2016, 8.3.2.

The structure of those fact hierarchies is defined in A.2, as a collection of MPEG-7 Classification Schemes, one for each fact hierarchy. Identified by the following URIs:

— `urn:mpeg:mpeg21:cel:ipre:2015/AccessPolicy`;

— `urn:mpeg:mpeg21:cel:ipre:2015/DeliveryModality`;

— `urn:mpeg:mpeg21:cel:ipre:2015/Device`;

— `urn:mpeg:mpeg21:cel:ipre:2015/Means`;

— `urn:mpeg:mpeg21:cel:ipre:2015/ServiceAccessPolicy`;

— `urn:mpeg:mpeg21:cel:ipre:2015/UserTimeAccess`.

The URI of each fact is thus defined by the adding the ‘termID’ as suffix of the Classification Scheme URI, separated by the ‘#’ character.

EXAMPLE 1 The URI of the constraint ‘`Linear`’ is ‘`urn:mpeg:mpeg21:cel:ipre:2015/DeliveryModality#Linear`’.

Notice that the URI of the single term does not reflect the position in the hierarchy.

When a deontic clause includes a constraint specified by a required fact with a hierarchy, the constraint is satisfied also by any facts which are child elements of such fact in the fact hierarchy.

EXAMPLE 2 A constraint requiring ‘Linear’ is in the domain of the ‘DeliveryModality’ and can be satisfied also by ‘Broadcasting’ or ‘Webcasting’.

This mechanism allows the parties to define constraints with the desired degree of generality/specificity.

In a few cases a fact falls under multiple sub-hierarchies (for example, ‘MobileBroadcastTechnology’) in the domain of ‘Means’, appears both under ‘BroadcastTechnology’ and ‘MobileTechnology’.

This may have implication in complex logical constructs for constraints.

EXAMPLE 3 In a clause requiring ‘BroadcastTechnology’ and also prohibiting ‘MobileTechnology’, ‘MobileBroadcastTechnology’ does not satisfy the constraint, while any other ‘BroadcastTechnology’ does.

The XML elements for the constraints defined as fact hierarchy and provided in the subsequent subclauses are all based on the generic complex type given below, where the required attribute `href` should contain the URI of the selected fact.

```
<complexType name="ExploitationConditionType" abstract="true">
   <complexContent>
      <extension base="cel-core:Fact">
         <attribute name="href" type="anyURI" use="required"/>
      </extension>
   </complexContent>
</complexType>
```

7.4.3.3 AccessPolicy

The `cel-ipre:AccessPolicy` constraint specifies if the user has to pay, or not, for content fruition, according to the suffix of the value given to its attribute `href` that should be one of the following:

- `FreeOfCharge`: no charge is due, by the final user for the content fruition, other than fees levied by government agencies;
- `Pay`: the content is accessible through payment. `Pay` can be further specified as follows:
  - `Subscription`: the content is accessible through a payment of a subscription fee;
  - `PayPerView`: the content is accessible through a payment on a per-exhibition basis;
  - `PayPerPackage`: the content is accessible through a payment on a per-package basis.

The definition of the cel-ipre:AccessPolicyType in the XML schema restricts the acceptable values of attribute href, without describing its hierarchy.

```
<complexType name="AccessPolicyType">
   <complexContent>
      <restriction base="cel-ipre:ExploitationConditionType">
         <attribute name="href">
            <simpleType>
               <restriction base="anyURI">
                  <enumeration value=
                  "urn:mpeg:mpeg21:cel:ipre:2015/AccessPolicy#FreeOfCharge"/>
                  <enumeration value=
                  "urn:mpeg:mpeg21:cel:ipre:2015/AccessPolicy#Pay"/>
                  <enumeration value=
                  "urn:mpeg:mpeg21:cel:ipre:2015/AccessPolicy#PayPerPackage"/>
                  <enumeration value=
                  "urn:mpeg:mpeg21:cel:ipre:2015/AccessPolicy#PayPerView"/>
                  <enumeration value=
                  "urn:mpeg:mpeg21:cel:ipre:2015/AccessPolicy#Subscription"/>
               </restriction>
            </simpleType>
         </attribute>
      </restriction>
   </complexContent>
</complexType>
```

The hierarchy of the cel-ipre:AccessPolicy constraint is defined in A.2 by the Classification Scheme quoted hereafter.

```
<mpeg7:ClassificationScheme uri="urn:mpeg:mpeg21:cel:ipre:2015/AccessPolicy">
   <mpeg7:Term termID="FreeOfCharge"/>
   <mpeg7:Term termID="Pay">
      <mpeg7:Term termID="PayPerPackage"/>
      <mpeg7:Term termID="PayPerView"/>
      <mpeg7:Term termID="Subscription"/>
   </mpeg7:Term>
</mpeg7:ClassificationScheme>
```

7.4.3.4 CopyrightExceptionFact

The cel-ipre:CopyrightExceptionFact constraint specifies copyright exceptions. They are used to define conditions to be met in order to be able to invoke some rights. Permissions from one user to another are not needed to invoke a CopyrightException. CopyrightExceptions are given based on the existence of the corresponding CopyrightExceptionFact.

7.4.3.5 DeliveryModality

The cel-ipre:DeliveryModality constraint restricts the communication to the public on who, and how, has the control on the time and place of the fruition by means of the href attribute whose suffix should take one of the following values.

- Linear: the media content is simultaneously delivered to end users by a service provider. Inside Linear one can have:
  - Broadcasting: the media content is delivered to end users by means of a point-to-multipoint technology;

  - `Webcasting`: the media content is delivered to end users by streaming via Internet.
- `NonLinear`: the media content is delivered to an end user as response to her individual request by a service provider. Inside it, one can have:
  - `OnDemandBasis`: the media content is delivered to an end user as response to an individual request. Inside it, one can have:
    - `OnDemandDownload`: the media content is delivered to the user system for one-time or endless reproduction;
    - `OnDemandStreaming`: the content is delivered in streaming modality as response to an individual request.

The definition of the `cel-ipre:DeliveryModalityType` in the XML schema does not allow describing its hierarchy.

```
<complexType name="DeliveryModalityType">
   <complexContent>
      <restriction base="cel-ipre:ExploitationConditionType">
         <attribute name="href">
            <simpleType>
                  <restriction base="anyURI">
                     <enumeration value=
                     "urn:mpeg:mpeg21:cel:ipre:2015/DeliveryModality#Linear"/>
                     <enumeration value=
                     "urn:mpeg:mpeg21:cel:ipre:2015/DeliveryModality#Broadcasting"/>
                     <enumeration value=
                     "urn:mpeg:mpeg21:cel:ipre:2015/DeliveryModality#Webcasting"/>
                     <enumeration value=
                     "urn:mpeg:mpeg21:cel:ipre:2015/DeliveryModality#NonLinear"/>
                     <enumeration value=
                     "urn:mpeg:mpeg21:cel:ipre:2015/DeliveryModality#OnDemandBasis"/>
                     <enumeration value=
                     "urn:mpeg:mpeg21:cel:ipre:2015/DeliveryModality#OnDemandDownload"/>
                     <enumeration value=
                     "urn:mpeg:mpeg21:cel:ipre:2015/DeliveryModality#OnDemandStreaming"/>
                  </restriction>
               </simpleType>
            </attribute>
         </restriction>
      </complexContent>
</complexType>
```

The hierarchy of the `cel-ipre:DeliveryModalityType` constraint is defined in A.2 by the Classification Scheme quoted hereafter.

```
<mpeg7:ClassificationScheme uri="urn:mpeg:mpeg21:cel:ipre:2015/DeliveryModality">
   <mpeg7:Term termID="Linear">
      <mpeg7:Term termID="Broadcasting"/>
      <mpeg7:Term termID="Webcasting"/>
   </mpeg7:Term>
   <mpeg7:Term termID="NonLinear">
      <mpeg7:Term termID="OnDemandBasis">
         <mpeg7:Term termID="OnDemandDownload"/>
         <mpeg7:Term termID="OnDemandStreaming"/>
      </mpeg7:Term>
   </mpeg7:Term>
</mpeg7:ClassificationScheme>
```

7.4.3.6 Device

The `cel-ipre:Device` constraint restricts the type of equipment used for the fruition of the content according to the suffix of the value given to its attribute `href`, whose suffix should be one of the following.

- `Computer`: the end user rendering device is a computer.
- `MobileDevice`: the end user rendering device is a portable equipment that can receive the media content in user motion. Inside it, one can have:
  - `MobileBroadcastDevice`: the end user rendering device is a mobile device that can receive the media content by means of wireless technologies;
  - `MobileTelecommunicationDevice`: the end user rendering device is a mobile device that can receive the media content by means of point-to-point mobile telecommunications technologies.
- `RobotDevice`: the end user rendering device is any equipment that can perform automated tasks without human interaction.
- `StorageDevice`: the end user rendering device is any equipment with storage functionalities.
- `TelevisionDevice`: the end user rendering device is any equipment with television functionalities. Inside it, one can have:
  - `TelevisionSet`: the end user rendering device is any equipment with television functionalities, but is not a mobile broadcast device.

The definition of the `cel-ipre:Device` in the XML schema does not allow describing its hierarchy.

```
<complexType name="DeviceType">
   <complexContent>
      <restriction base="cel-ipre:ExploitationConditionType">
         <attribute name="href">
            <simpleType>
               <restriction base="anyURI">
                  <enumeration value=
                  "urn:mpeg:mpeg21:cel:ipre:2015/Device#Computer"/>
                  <enumeration value=
                  "urn:mpeg:mpeg21:cel:ipre:2015/Device#MobileDevice"/>
                  <enumeration value=
                  "urn:mpeg:mpeg21:cel:ipre:2015/Device#MobileBroadcastDevice"/>
                  <enumeration value=
                  "urn:mpeg:mpeg21:cel:ipre:2015/Device#MobileTelecommunicationDevice"/>
                  <enumeration value=
                  "urn:mpeg:mpeg21:cel:ipre:2015/Device#RobotDevice"/>
                  <enumeration value=
                  "urn:mpeg:mpeg21:cel:ipre:2015/Device#StorageDevice"/>
                  <enumeration value=
                  "urn:mpeg:mpeg21:cel:ipre:2015/Device#TelevisionDevice"/>
                  <enumeration value=
                  "urn:mpeg:mpeg21:cel:ipre:2015/Device#TelevisionSet"/>
               </restriction>
            </simpleType>
         </attribute>
      </restriction>
   </complexContent>
</complexType>
```

The hierarchy of the cel-ipre:Device constraint is defined in A.2 by the Classification Scheme quoted hereafter.

```
<mpeg7:ClassificationScheme uri="urn:mpeg:mpeg21:cel:ipre:2015/Device">
   <mpeg7:Term termID="Computer"/>
   <mpeg7:Term termID="MobileDevice">
      <mpeg7:Term termID="MobileBroadcastDevice"/>
      <mpeg7:Term termID="MobileTelecommunicationDevice"/>
   </mpeg7:Term>
   <mpeg7:Term termID="RobotDevice"/>
   <mpeg7:Term termID="StorageDevice"/>
   <mpeg7:Term termID="TelevisionDevice">
      <mpeg7:Term termID="MobileBroadcastDevice"/>
      <mpeg7:Term termID="TelevisionSet"/>
   </mpeg7:Term>
</mpeg7:ClassificationScheme>
```

7.4.3.7 Means

The cel-ipre:Means constraint restricts the way in which the content is delivered to the public by means of the href attribute whose suffix should take one of the following values.

- Videogram: the technology used for delivering media content support the creation of a media content physical container in any format (e.g. videocassettes, DVD, DVD-RIM, Internet access ready DVD, CDi, CD, CD-ROM, UMD, VCD).
- TransmissionTechnology: any means based on transmission technology, which can be further specifed as follows:
  - BroadcastTechnology: the technology used for delivering media content is point-to-multipoint. Inside it, one can have:
    - Cable: the technology used for delivering media content is co-axial and/or fiber optic cable (excluding DSL, ADSL, Internet or other IP-based networks) and it will be rendered on television set;
    - IPNetwork: the technology used for delivering media content is DSL, ADSL or any IP-based network via a private network (excluding Internet);
    - MobileBroadcastTechnology: the technology used for delivering media content is point-to-multi-point and the content will be rendered on portable devices;
    - Satellite: the technology used for delivering media content makes use of a geostationary satellite system;
    - Terrestrial: the technology used for delivering media content makes use of a terrestrial television transmitter.
  - Internet: the technology used for delivering media content is based on TCP/IP software protocols (or equivalent).
  - MobileTechnology: the technology used for delivering media content is suitable for rendering in a mobile device. Inside it, one can have:
    - MobileTelecommunicationTechnology: the technology used for delivering media content is point-to-point and the content will be rendered on portable devices;
    - MobileBroadcastTechnology: the same as above (for double inheritance).

The definition of the cel-ipre:MeansType in the XML schema does not allow describing its hierarchy.

```
<complexType name="MeansType">
   <complexContent>
      <restriction base="cel-ipre:ExploitationConditionType">
         <attribute name="href">
            <simpleType>
               <restriction base="anyURI">
                  <enumeration value=
                  "urn:mpeg:mpeg21:cel:ipre:2015/Means#Videogram"/>
                  <enumeration value=
                  "urn:mpeg:mpeg21:cel:ipre:2015/Means#TransmissionTechnology"/>
                  <enumeration value=
                  "urn:mpeg:mpeg21:cel:ipre:2015/Means#BroadcastTechnology"/>
                  <enumeration value=
                  "urn:mpeg:mpeg21:cel:ipre:2015/Means#Cable"/>
                  <enumeration value=
                  "urn:mpeg:mpeg21:cel:ipre:2015/Means#IPNetwork"/>
                  <enumeration value=
                  "urn:mpeg:mpeg21:cel:ipre:2015/Means#MobileBroadcastTechnology"/>
                  <enumeration value=
                  "urn:mpeg:mpeg21:cel:ipre:2015/Means#Satellite"/>
                  <enumeration value=
                  "urn:mpeg:mpeg21:cel:ipre:2015/Means#Terrestrial"/>
                  <enumeration value=
                  "urn:mpeg:mpeg21:cel:ipre:2015/Means#Internet"/>
                  <enumeration value=
                  "urn:mpeg:mpeg21:cel:ipre:2015/Means#MobileTechnology"/>
                  <enumeration value=
                  "urn:mpeg:mpeg21:cel:ipre:2015/Means#MobileTelecommunicationTechnolo-
gy"/>
               </restriction>
            </simpleType>
         </attribute>
      </restriction>
   </complexContent>
</complexType>
```

The hierarchy of the cel-ipre:Means constraint is defined in A.2 by the Classification Scheme quoted hereafter.

```
<mpeg7:ClassificationScheme uri="urn:mpeg:mpeg21:cel:ipre:2015/Means">
   <mpeg7:Term termID="TransmissionTechnology">
      <mpeg7:Term termID="BroadcastTechnology">
         <mpeg7:Term termID="Cable"/>
         <mpeg7:Term termID="IPNetwork"/>
         <mpeg7:Term termID="MobileBroadcastTechnology"/>
         <mpeg7:Term termID="Satellite"/>
         <mpeg7:Term termID="Terrestrial"/>
      </mpeg7:Term>
      <mpeg7:Term termID="Internet"/>
      <mpeg7:Term termID="MobileTechnology">
         <mpeg7:Term termID="MobileBroadcastTechnology"/>
         <mpeg7:Term termID="MobileTelecommunicationTechnology"/>
      </mpeg7:Term>
   </mpeg7:Term>
   <mpeg7:Term termID="Videogram"/>
</mpeg7:ClassificationScheme>
```

7.4.3.8 ServiceAccessPolicy

The cel-ipre:ServiceAccessPolicy constraint restricts the access to services according to the suffix of the value given to its attribute href that should be one of the following.

- Open: the service can be provided to end users without the need for approval of the service provider.

- Restricted: the service only can be provided to end users with the approval of the service provider. Inside it, one can have:
  - Hotel models the restriction that the service is provided only to the customers of permanent or temporary living accommodations, such as hotels, motels, or apartments, within the area of such accommodations or in their immediate neighborhood;
  - PublicPerformanceHalls models the restriction that an cel-ipre:PublicPerformance is executed in places, such as Cinemas or Theatres, specifically conceived and authorized for public performances;
  - Transportations models the restriction that the service is provided only to the customers of transportation services, during their fruition on board. Transportations include, among others, the following values:
    - Airplanes models the restriction that the service is provided only to the customers of airplanes transportation services, during their fruition on board;
    - BusesMetro models the restriction that the service is provided only to the customers of urban public transportation services, including buses, trams, and metro (or rapid transit), during their fruition on board;
    - Ships models the restriction that the service is provided only to the customers of ship transportation services, during their fruition on board;
    - Trains models the restriction that the service is provided only to the customers of train transportation services, during their fruition on board.

The definition of the cel-ipre:ServiceAccessPolicyType in the XML schema does not allow describing its hierarchy.

```
<complexType name="ServiceAccessPolicyType">
   <complexContent>
      <restriction base="cel-ipre:ExploitationConditionType">
         <attribute name="href">
         <simpleType>
               <restriction base="anyURI">
                     <enumeration value=
                     "urn:mpeg:mpeg21:cel:ipre:2015/ServiceAccessPolicy#Open"/>
                     <enumeration value=
                     "urn:mpeg:mpeg21:cel:ipre:2015/ServiceAccessPolicy#Restricted"/>
                     <enumeration value=
                     "urn:mpeg:mpeg21:cel:ipre:2015/ServiceAccessPolicy#Hotel"/>
                     <enumeration value=
                     "urn:mpeg:mpeg21:cel:ipre:2015/ServiceAccessPolicy#PublicPerfomance-
Halls"/>
                     <enumeration value=
                     "urn:mpeg:mpeg21:cel:ipre:2015/ServiceAccessPolicy#Transporta-
tions"/>
                     <enumeration value=
                     "urn:mpeg:mpeg21:cel:ipre:2015/ServiceAccessPolicy#Airplanes"/>
                     <enumeration value=
                     "urn:mpeg:mpeg21:cel:ipre:2015/ServiceAccessPolicy#BusesMetro"/>
                     <enumeration value=
                     "urn:mpeg:mpeg21:cel:ipre:2015/ServiceAccessPolicy#Ships"/>
                     <enumeration value=
                     "urn:mpeg:mpeg21:cel:ipre:2015/ServiceAccessPolicy#Trains"/>
               </restriction>
            </simpleType>
         </attribute>
      </restriction>
   </complexContent>
</complexType>
```

The hierarchy of the `cel-ipre:ServiceAccessPolicy` constraint is defined in A.2 by the Classification Scheme quoted hereafter.

```
<mpeg7:ClassificationScheme uri="urn:mpeg:mpeg21:cel:ipre:2015/ServiceAccessPolicy">
   <mpeg7:Term termID="Open"/>
   <mpeg7:Term termID="Restricted">
      <mpeg7:Term termID="Hotel"/>
      <mpeg7:Term termID="PublicPerformanceHalls"/>
      <mpeg7:Term termID="Transportations">
            <mpeg7:Term termID="Airplanes"/>
            <mpeg7:Term termID="BusesMetro"/>
            <mpeg7:Term termID="Ships"/>
            <mpeg7:Term termID="Trains"/>
      </mpeg7:Term>
   </mpeg7:Term>
</mpeg7:ClassificationScheme>
```

7.4.3.9 UserTimeAccess

The `cel-ipre:UserTimeAccess` constraint restricts the time of availability of the content for final user fruition. If the time is limited, it has the child element `cel-ipre:Limited`, and optionally, the validity period can be expressed by means of the validity attribute that determines how long the user can access after the first access. Otherwise, it has the `cel-ipre:Unlimited` child element.

Figure 31 shows the `cel-ipre:UserTimeAccess` element structure.

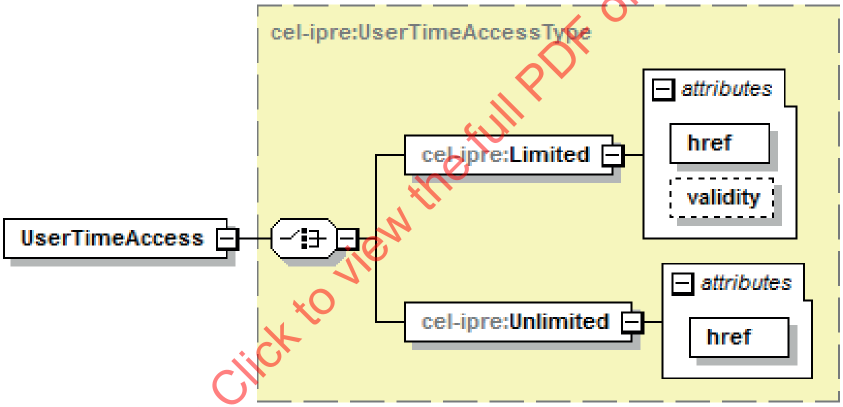

Figure 31 — UserTimeAccess constraint

```
<complexType name="UserTimeAccessType">
   <complexContent>
      <extension base="cel-core:Fact">
         <choice>
            <element name="Limited">
               <complexType>
                  <attribute name="href" type="anyURI"
                  fixed="urn:mpeg:mpeg21:cel:ipre:2015/UserTimeAccess#Limited"/>
                  <attribute name="hasValidity" type="duration"/>
               </complexType>
            </element>
            <element name="Unlimited">
               <complexType>
                  <attribute name="href" type="anyURI"
                  fixed="urn:mpeg:mpeg21:cel:ipre:2015/UserTimeAccess#Unlimited"/>
               </complexType>
            </element>
         </choice>
      </extension>
   </complexContent>
</complexType>
```

7.4.3.10 TemporalContext

The `cel-ipre:TemporalContext` constraint restricts the time period of execution of the action specified in the deontic clause. It consists of the following two attributes, as shown in Figure 32:

— `afterDate`: date after which the action can/shall/cannot be executed;

— `beforeDate`: date before which the action can/shall/cannot be executed.

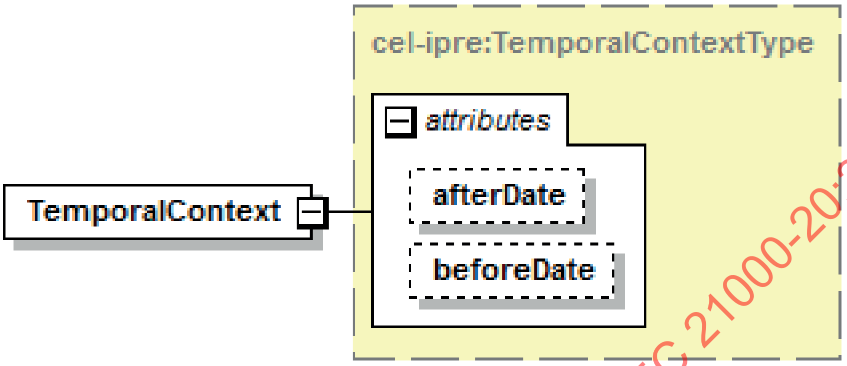

Figure 32 — TemporalContext constraint

```
<complexType name="TemporalContextType">
   <complexContent>
      <extension base="cel-core:Fact">
         <attribute name="afterDate" type="dateTime"/>
         <attribute name="beforeDate" type="dateTime"/>
      </extension>
   </complexContent>
</complexType>
```

7.4.3.11 SpatialContext

The `cel-ipre:SpatialContext` constraint restricts the territory on which the action can be executed. It consists of the `cel-ipre:Country` and `cel-ipre:Region` elements which specify the country(s) and/or region(s) where the action can be executed. The countries can be represented by means of the codes in ISO 3166-1.

Figure 33 shows the `cel-ipre:SpatialContext` element structure.

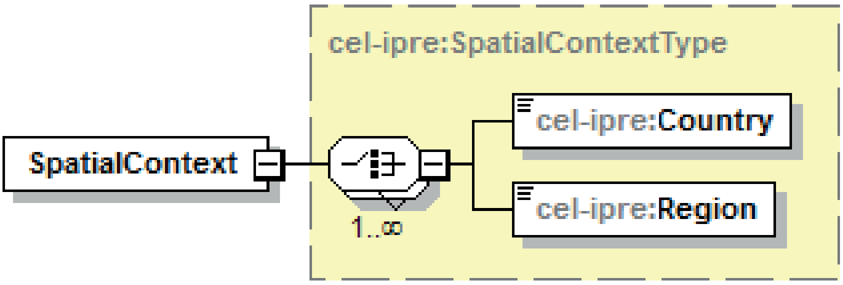

Figure 33 — SpatialContext constraint

```
<complexType name="SpatialContextType">
   <complexContent>
      <extension base="cel-core:Fact">
         <choice maxOccurs="unbounded">
            <element name="Country" type="string"/>
            <element name="Region" type="string"/>
         </choice>
      </extension>
   </complexContent>
</complexType>
```

7.4.3.12 Runs

The `cel-ipre:Runs` constraint restricts the maximum number of executions of the action in a deontic expression, as well as the period of validity. It has three attributes, as depicted in Figure 34:

- `numberOfRuns`: maximum number of allowed runs;
- `validity`: attributes the period during which an unlimited or specified number of repetitions have to be considered as a single run; it is optional and associated to `numberOfRepetitions`;
- `numberOfRepetitions`: specifies the number of repetitions (within a validity window) to be considered a single run. It is optional and requires that validity is given.

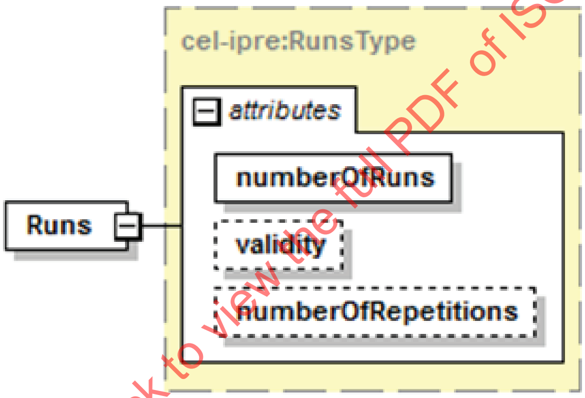

Figure 34 — Runs constraint

```
<complexType name="RunsType">
   <complexContent>
      <extension base="cel-core:Fact">
         <attribute name="numberOfRuns" type="nonNegativeInteger" use="required"/>
         <attribute name="validity" type="duration" use="optional"/>
         <attribute name="numberOfRepetitions" type="nonNegativeInteger" use="optional"/>
      </extension>
   </complexContent>
</complexType>
```

7.4.3.13 Language

The `cel-ipre:Language` constraint states the languages permitted/obligated/prohibited for public communication. It has the languages attribute as shown in Figure 35, in which the languages are specified, for example, by means of an ISO code as defined in ISO 639-1.

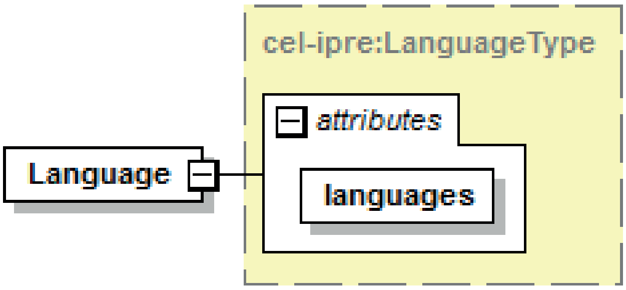

Figure 35 — Language constraint

```
<complexType name="LanguageType">
   <complexContent>
      <extension base="cel-core:Fact">
         <attribute name="languages" use="required">
            <simpleType>
               <list itemType="string"/>
            </simpleType>
         </attribute>
      </extension>
   </complexContent>
</complexType>
```

7.4.3.14 Length

The cel-ipre:Length constraint restricts the length in terms of duration (canonical play) by means of its attribute maxLength which specifies the maximum length of an object resultant from an action. The maxLength attribute has type xsd:duration. Figure 36 shows the structure of the cel-ipre:Length element.

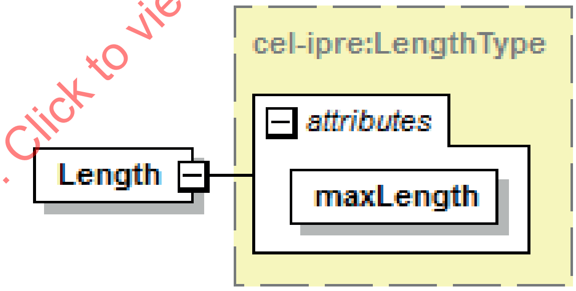

Figure 36 — Length constraint

```
<complexType name="LengthType">
   <complexContent>
      <extension base="cel-core:Fact">
         <attribute name="maxLength" type="duration" use="required"/>
      </extension>
   </complexContent>
</complexType>
```

7.4.3.15 IPEntityContext

The cel-ipre:IPEntityContext constraint restricts the cel-core:Object in a deontic expression to be part of another object identified by the dii:Identifier element or by the dii:RelatedIdentifier element of its child element cel-ipre:PartOf. Figure 37 shows the structure of the cel-ipre:IPEntityContext element.

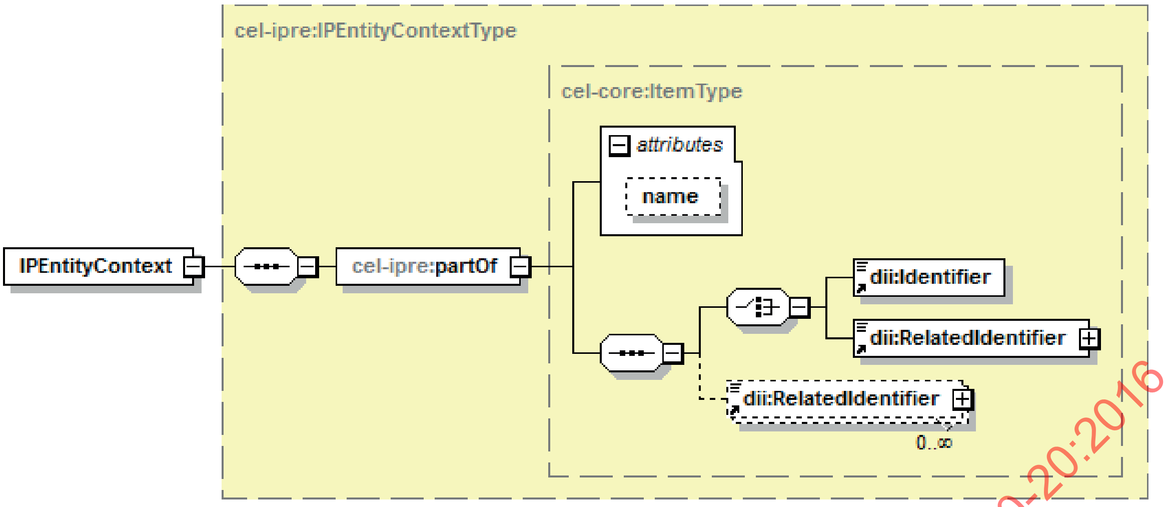

Figure 37 — IPEntityContext constraint

```
<complexType name="IPEntityContextType">
   <complexContent>
      <extension base="cel-core:Fact">
         <sequence>
            <element name="PartOf" type="cel-core:ItemType"/>
         </sequence>
      </extension>
   </complexContent>
</complexType>
```

7.4.3.16 ServiceChannelContext

The `cel-ipre:ServiceChannelContext` models the restriction on the outlets used for the exploitation of the rights (e.g. the communication to the public); such outlets, recognizable, can be indifferently named as Services (possibly with a common brand) or Channels, according to the context. They can be listed in the `servicesAndChannels` attribute.

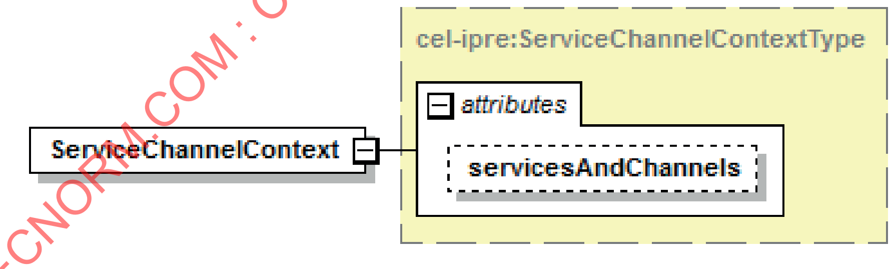

Figure 38 — ServiceChannelContext constraint

```
<complexType name="ServiceChannelContextType">
   <complexContent>
      <extension base="cel-core:Fact">
         <attribute name="servicesAndChannels">
            <simpleType>
               <list itemType="anyURI"/>
            </simpleType>
         </attribute>
      </extension>
   </complexContent>
</complexType>
```

7.4.3.17 MaterialFormat

The `cel-ipre:MaterialFormat` models the restrictions on the technical properties of the material (copies or instances) used for, or resulting from, the exploitation of the rights (e.g. the communication to the public). See Figure 39.

This element has the following elements.

- `cel-ipre:MatchesFormatComplianceProfile` — it indicates which is the format compliance profile that defines the itemized technical properties, with their expected values and (when relevant) their assessment thresholds, against which the constrained material format shall match.
- `cel-ipre:Format` — it describes which format is permitted (or obligated or forbidden) when the material format is constrained.
- `cel-ipre:VideoFormat` — it describes which video format is permitted (or obligated or forbidden) when the material format is constrained.
- `cel-ipre:AudioFormat` — it describes which audio format is permitted (or obligated or forbidden) when the material format is constrained.
- `cel-ipre:MaxBitrate` — it indicates the maximum bitrate, in kilobits per second, for the constrained material.
- `cel-ipre:MinBitrate` — it indicates the minimum bitrate, in kilobits per second, for the constrained material.
- `cel-ipre:AspectRatio` — it describes which aspect ratios are permitted (or obligated or forbidden) for the constrained material.
- `cel-ipre:MaxLines` — it indicates the maximum number of picture lines of the constrained material.
- `cel-ipre:MinLines` — it indicates the minimum number of picture lines of the constrained material.

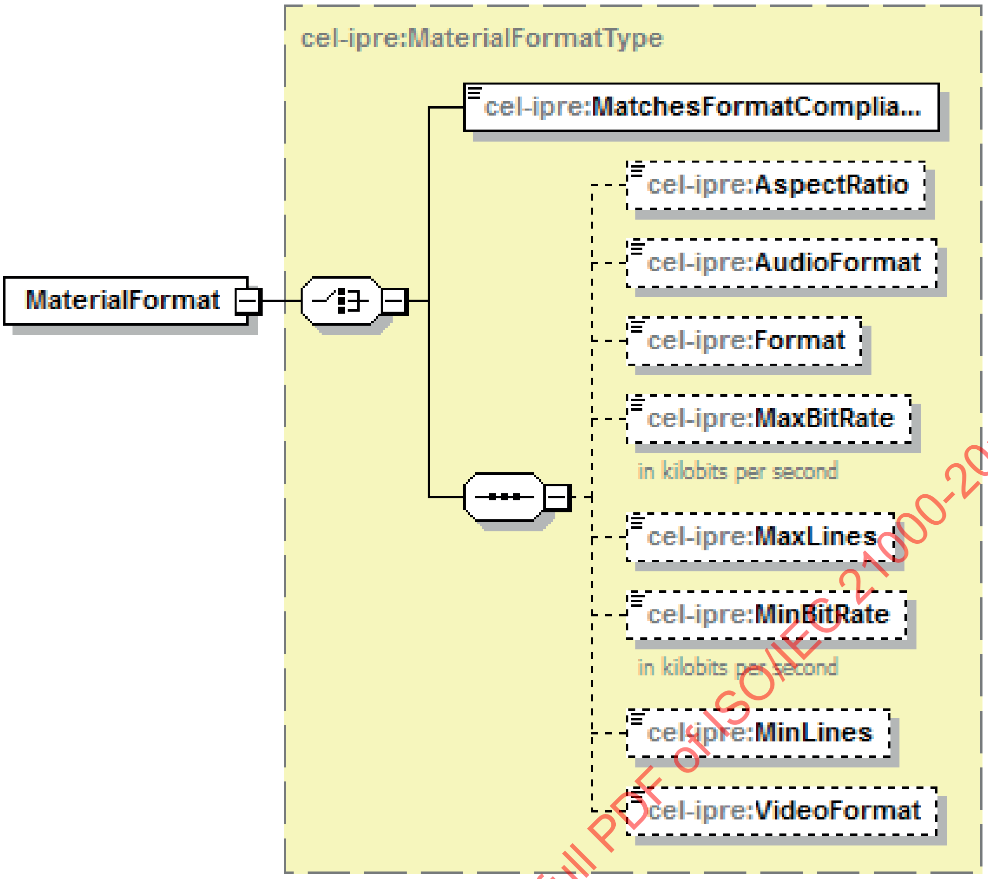

Figure 39 — MaterialFormat constraint

```
<complexType name="MaterialFormatType">
   <complexContent>
      <extension base="cel-core:Fact">
         <choice>
            <element name="MatchesFormatComplianceProfile" type="anyURI"/>
            <sequence>
               <element name="AspectRatio" minOccurs="0">
                  <simpleType>
                     <list itemType="string"/>
                  </simpleType>
               </element>
               <element name="AudioFormat" type="string" minOccurs="0"/>
               <element name="Format" type="string" minOccurs="0"/>
               <element name="MaxBitRate" type="positiveInteger" minOccurs="0">
                  <annotation>
                     <documentation>in kilobits per second</documentation>

                  </annotation>
               </element>
               <element name="MaxLines" type="positiveInteger" minOccurs="0"/>
               <element name="MinBitRate" type="positiveInteger" minOccurs="0">
                  <annotation>
                     <documentation>in kilobits per second</documentation>
                  </annotation>
               </element>
               <element name="MinLines" type="positiveInteger" minOccurs="0"/>
               <element name="VideoFormat" type="string" minOccurs="0"/>
            </sequence>
         </choice>
      </extension>
   </complexContent>
</complexType>
```

7.4.4 Permission attributes

The cel-core:Permission element requires some specific attributes when dealing with intellectual property rights, as depicted in Figure 40.

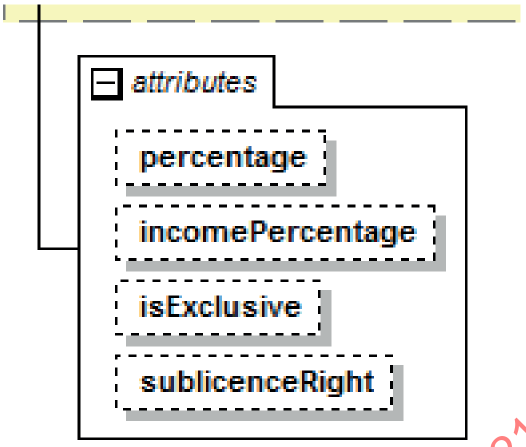

Figure 40 — Permission element

Then, a cel-ipre:Permission element is defined with the following attributes.

- percentage — indicates when the permission is shared with other actors (not 100 %) and thus the permission cannot be actually exploited. If the permission does not expire, i.e. it does not require a cel-ipre:TemporalContext with beforeDate attribute, it can be considered a "percentage of ownership"; otherwise, it shall be considered only as a "percentage of use". It is of type xsd:decimal.
- incomePercentage — indicates when the income of the exploitation of the permission has to be shared. It is of type xsd:decimal.
- isExclusive — indicates if this exploitation might be granted by the issuer to multiple licensees in the same context or not. If not explicitly given, the exclusivity shall not be considered granted. Attention should be paid that, in the case of exclusivity, if cel-ipre:hasSublicenseRight=false, then the granted exploitation could not be object of further trade. It is of type xsd:boolean.
- sublicenceRight —indicates if it is possible to sublicense a granted right (a permission on the permission). If not explicitly given, the permission to sublicense can be considered as granted. It is of type xsd:boolean.

7.5 CEL Extension for payments and notifications

7.5.1 General

The CEL extension for payments and notifications fully confirms the contract model defined by the Media Contract Core defined in cel-core.

Besides this extension does not conflict with the CEL extension for the Exploitation of Intellectual Property Rights, defined in cel-ipre.

This subclause systematically describes the schema of the CEL extension for payments and notifications in the domain of Media Contracts.

The URI of CEL extension for payments and notifications is:

```
urn:mpeg:mpeg21:cel:pane:2015
```

and the prefix used in this document is:

cel-pane

7.5.2 Payment

The cel-pane:Payment element derives from the cel-core:Action element and it models a payment action between contract parties (see Figure 41). The cel-pane:Payment involves two parties of the contract, with different roles: one acting the payment and the other one being its beneficiary. The payment is fully specified by the data properties giving the amount and the currency. It has to be defined inside a cel-core:Obligation element.

The following attributes are defined for the element cel-pane:Payment:

— amount — with range xsd:decimal, it attributes the precise value of the payment amount, when it is defined in the contract;

— currency — it attributes the currency of the payment, with expected values as three character string according to ISO 4217, not strictly enforced;

— incomePercentage — with range xsd:decimal, it attributes the payment amount when it is defined as a percentage on the net income resulting from a rights exploitation action; in this case, the amount of the payment cannot be known in advance.

The cel-pane:Payment includes two elements:

— cel-pane:Beneficiary with the required attribute partyRef of type IDREF which models the relation used to express the recipient Party for a Payment action;

— cel-pane:IncomeSource when the payment amount is defined as a percentage on the net income resulting from a rights exploitation action, to indicate that action directly.

The following attribute is defined for cel-pane:Beneficiary:

— partyRef — it contains the ID of the referenced cel-core:Party element.

The following attribute is defined for cel-pane:IncomeSource:

— actRef — it contains the ID of the referenced cel-core:Act element.

Each payment is assumed to occur once, after which the obligation is satisfied. It is possible to define the time period of validity of the obligation by means of a cel-ipre:TemporalContext constraint, in which data property beforeDate specifies the deadline, while afterDate can be used to specify the start of the acceptance period for the payment.

An obligation to payment can also be defined in terms of a percentage over the net income resulting from an exploitation action. In this case, the amount of the payment cannot be known in advance. Moreover, the obligation is expected to get valid on the accomplishment of the related action.

Notice that, as each payment action can occur only once, the parties have to define one obligation for each payment occurrence they need to agree, such as instalments or repeated payments. This implies that the definition of an unbounded number of payments is not supported.

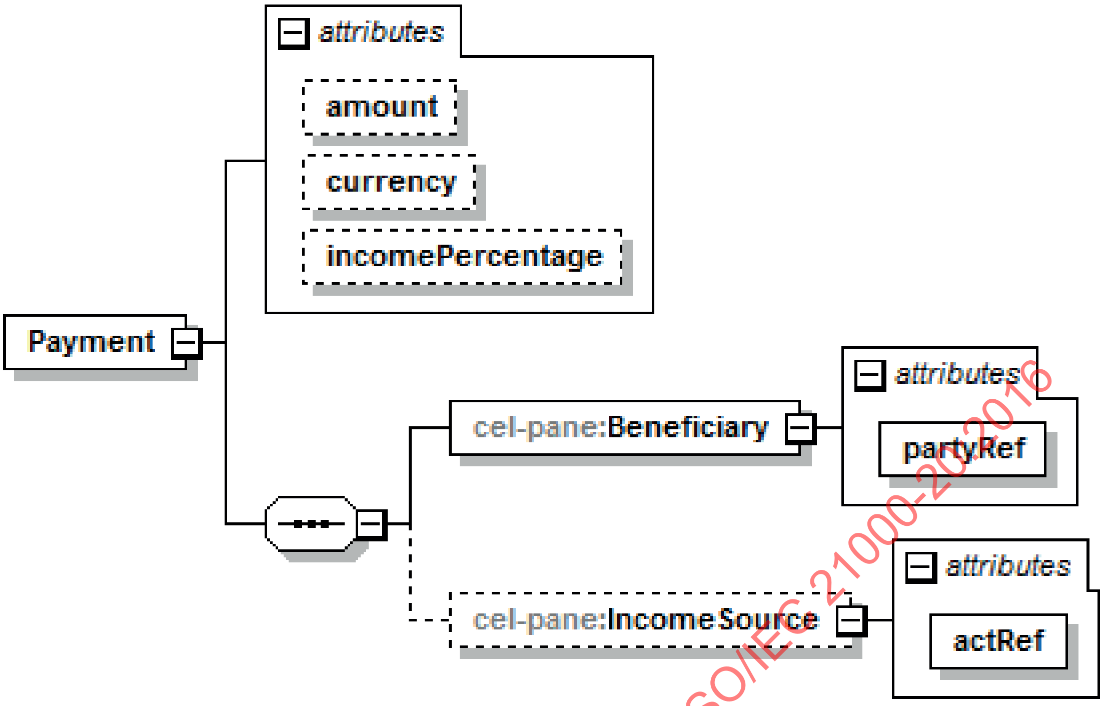

Figure 41 — Payment element

An example is given in the grey box below, according to schema fragment given in the orange box.

```
<element name="Payment" substitutionGroup="cel-core:Action">
   <complexType>
      <complexContent>
         <extension base="cel-core:Action">
            <sequence>
               <element name="Beneficiary">
                  <complexType>
                     <attribute name="partyRef" type="IDREF" use="required"/>
                  </complexType>
               </element>
               <element name="IncomeSource" minOccurs="0">
                  <complexType>
                     <attribute name="actRef" type="IDREF" use="required"/>
                  </complexType>
               </element>
            </sequence>
            <attribute name="amount" type="decimal"/>
            <attribute name="currency" type="string"/>
            <attribute name="incomePercentage" type="decimal"/>
         </extension>
      </complexContent>
   </complexType>
</element>
```

```
   <cel-core:Obligation id="x2">
      <cel-core:Subject partyRef="PartyA"/>
      <cel-core:Act>
         <cel-pane:Payment amount="199000" currency="EUR">
            <cel-pane:Beneficiary partyRef="PartyB"/>
         </cel-pane:Payment>
      </cel-core:Act>
   </cel-core:Obligation>
```

7.5.3 Notify

The `cel-pane:Notify` element derives from the `cel-core:Action` element and it models a notification to a party (see Figure 42). The notification involves two parties of the contract, with different roles: one acting the notification and the other one being its recipient.

The `cel-pane:Notify` includes two elements:

— `cel-pane:Recipient` — which is used to express the recipient Party for a notification action;

— `cel-pane:About` — which is used to express directly which action is the object of the notification.

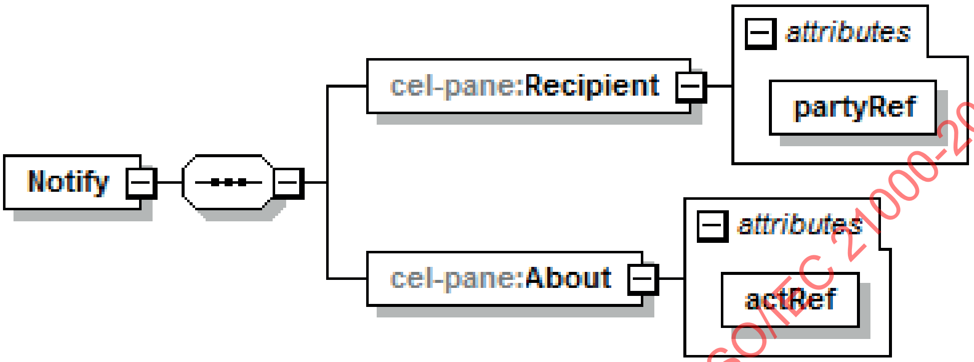

Figure 42 — Notify element

The following attribute is defined for `cel-pane:Recipient`:

— `partyRef` — it contains the ID of the referenced `cel-core:Party` element.

The following attribute is defined for `cel-pane:About`:

— `actRef` — it contains the ID of the referenced `cel-core:Act` element.

A notification is intended to report about the occurrence of a related action, which can be specified through a specific relation (`cel-pane:about`).

```
<element name="Notify" substitutionGroup="cel-core:Action">
   <complexType>
      <complexContent>
         <extension base="cel-core:Action">
            <sequence>
               <element name="Recipient">
                  <complexType>
                     <attribute name="partyRef" type="IDREF" use="required"/>
                  </complexType>
               </element>
               <element name="About">
                  <complexType>
                     <attribute name="actRef" type="IDREF" use="required"/>
                  </complexType>
               </element>
            </sequence>
         </extension>
      </complexContent>
   </complexType>
</element>
```

7.6 CEL Extension for Rights Expression Language standard and multimedia extensions rights and conditions

7.6.1 General

This subclause systematically describes the schema of the CEL extension for including rights and conditions coming from the standard and multimedia extensions of ISO/IEC 21000-5.

The URI of CEL extension for rights expression language standard and multimedia extensions rights and conditions is:

```
urn:mpeg:mpeg21:cel:rele:2015
```

and the prefix used in this document is:

```
cel-rele
```

7.6.2 Acts

The acts defined in this extension are the rights defined in standard and multimedia extensions of ISO/IEC 21000-5. The relationship between these rights and `cel-core` acts is depicted in Figure 43.

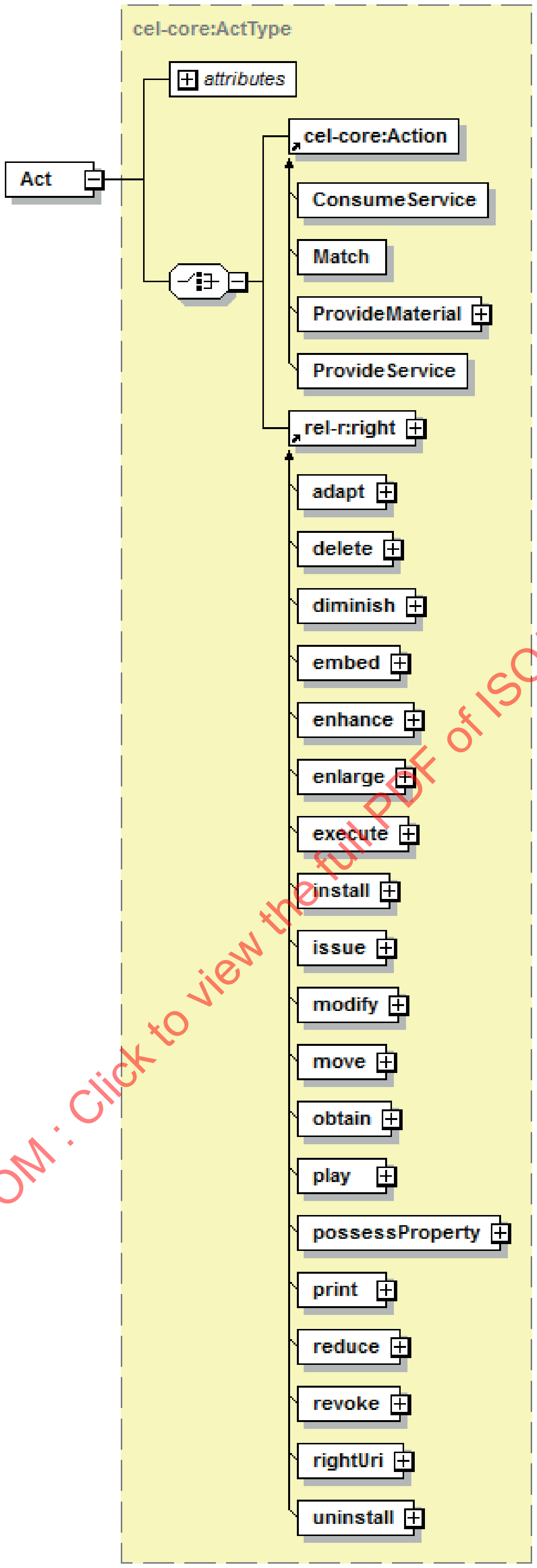

Figure 43 — Rights defined in standard and multimedia extensions

7.6.3 Constraints

The constraints defined in this extension are the conditions defined in standard and multimedia extensions of ISO/IEC 21000-5. The relationship between these conditions and `cel-core` constraints is depicted in Figure 44.

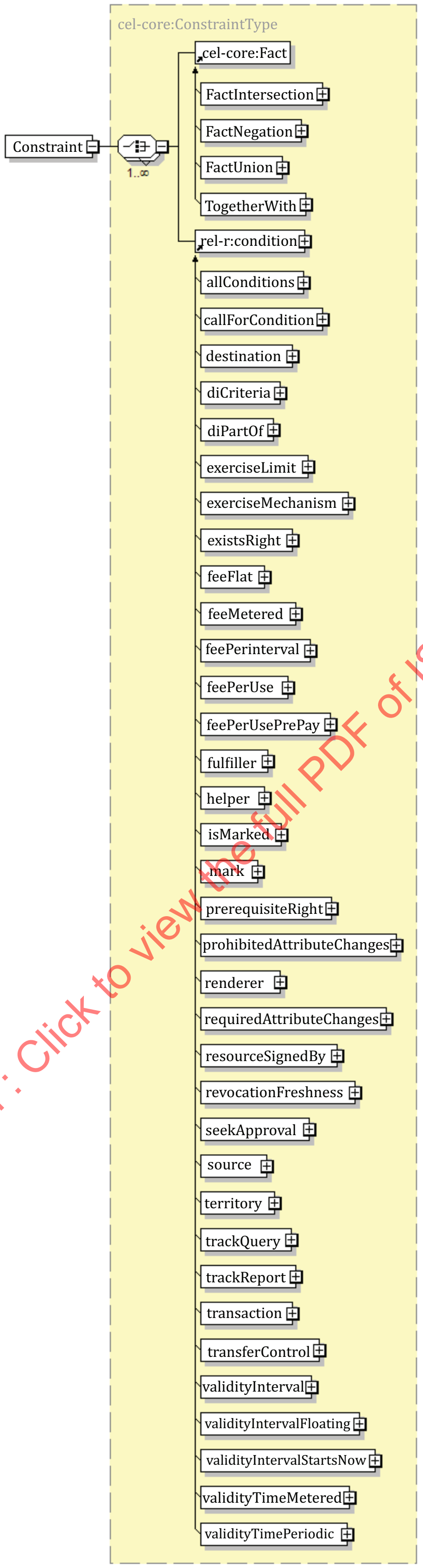

Figure 44 — Conditions defined in standard and multimedia extensions

7.7 Examples

7.7.1 General

This subclause provides the XML representation for six sample contracts, which include a simple permission, a simple prohibition, a simple obligation, a statement, a precondition and a complex permission.

7.7.2 Simple permission

This subclause provides the XML representation of a simple permission where User0002 is allowed to make an adaptation over the work Work001.

A permission is expressed by means of the cel-core:Permission element.

```
<cel-core:Contract>
    <cel-core:Metadata>
        <cel-core:SimpleDC>
           <dc:title>Example Permission p001</dc:title>
           <dc:date>2012-07-02</dc:date>
        </cel-core:SimpleDC>
    </cel-core:Metadata>
    <cel-core:Party id="user001">
       <cel-core:Person>
          <cel-core:Name>user001</cel-core:Name>
       </cel-core:Person>
    </cel-core:Party>
    <cel-core:Party id="user002">
       <cel-core:Person>
          <cel-core:Name>user002</cel-core:Name>
       </cel-core:Person>
    </cel-core:Party>
    <cel-core:Body>
       <cel-core:OperativePart>
          <cel-core:Permission id="p001">
             <cel-core:Subject partyRef="user002"/>
             <cel-core:Act>
                <cel-ipre:MakeAdaptation/>
             </cel-core:Act>
             <cel-core:Object>
                <cel-core:Item>
                   <dii:Identifier>work001</dii:Identifier>
                </cel-core:Item>
             </cel-core:Object>
             <cel-core:Issuer partyRef="user001"/>
          </cel-core:Permission>
       </cel-core:OperativePart>
    </cel-core:Body>
</cel-core:Contract>
```

7.7.3 Simple prohibition

This subclause provides the XML representation of a simple prohibition where User0002 is prohibited to make a copy over the work Work001. This prohibition has required the copyright exception fact.

A prohibition is expressed by means of the `cel-core:Prohibition` element.

```
<cel-core:Contract>
    <cel-core:Metadata>
       <cel-core:SimpleDC>
           <dc:title>Example Prohibition p002</dc:title>
           <dc:date>2012-07-02</dc:date>
        </cel-core:SimpleDC>
    </cel-core:Metadata>
    <cel-core:Party id="user001">
        <cel-core:Person>
           <cel-core:Name>user001</cel-core:Name>
        </cel-core:Person>
    </cel-core:Party>
    <cel-core:Party id="user002">
        <cel-core:Person>
           <cel-core:Name>user002</cel-core:Name>
        </cel-core:Person>
     </cel-core:Party>
    <cel-core:Body>
        <cel-core:OperativePart>
           <cel-core:Prohibition id="p002">
              <cel-core:Subject partyRef="user002"/>
              <cel-core:Act>
                 <cel-ipre:MakeCopy/>
              </cel-core:Act>
              <cel-core:Object>
                 <cel-core:Item>
                    <dii:Identifier>work001</dii:Identifier>
                 </cel-core:Item>
              </cel-core:Object>
              <cel-core:Constraint>
                 <cel-ipre:CopyrightExceptionFact/>
              </cel-core:Constraint>
              <cel-core:Issuer partyRef="user001"/>
           </cel-core:Prohibition>
        </cel-core:OperativePart>
     </cel-core:Body>
</cel-core:Contract>
```

7.7.4 Simple obligation

This subclause provides the XML representation of a simple obligation where `User0002` is obligated to distribute over the work `Work001` in the temporal interval (from 2011-07-22 to 2011-11-30).