INTERNATIONAL STANDARD

ISO/IEC 21000-20

First edition
2013-07-01

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 21000-20:2013(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

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.

ISO/IEC 21000-20 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

ISO/IEC 21000 consists of the following parts, under the general title *Information technology — Multimedia framework (MPEG-21)*:

- *Part 1: Vision, Technologies and Strategy* [Technical Report]
- *Part 2: Digital Item Declaration*
- *Part 3: Digital Item Identification*
- *Part 4: Intellectual Property Management and Protection Components*
- *Part 5: Rights Expression Language*
- *Part 6: Rights Data Dictionary*
- *Part 7: Digital Item Adaptation*
- *Part 8: Reference software*
- *Part 9: File Format*
- *Part 10: Digital Item Processing*
- *Part 11: Evaluation Tools for Persistent Association Technologies* [Technical Report]
- *Part 12: Test Bed for MPEG-21 Resource Delivery* [Technical Report]
- *Part 14: Conformance Testing*
- *Part 15: Event Reporting*
- *Part 16: Binary Format*

- *Part 17: Fragment Identification of MPEG Resources*
- *Part 18: Digital Item Streaming*
- *Part 19: Media Value Chain Ontology*
- *Part 20: Contract Expression Language*
- *Part 21: Media Contract Ontology*

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 enabled open market. This will also be to the benefit of the 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 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 21000-5. 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 is the ISO/IEC language for expressing such contracts in a structured representation.

Information technology — Multimedia framework (MPEG-21) —

Part 20: Contract Expression Language

1 Scope

This part of ISO/IEC 21000 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.

The Contract Expression Language (CEL) aims at digitally representing the agreements made in an environment of ISO/IEC 21000 use. These contracts include those about both transactions of content packed as Digital Items as well as services provided around this content.

The range of contracts under scope are:

— contracts about transactions of content as MPEG-21 Digital Items;

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

The aspects represented by CEL contracts include:

— 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 referenced documents are indispensable for the application 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 3166-1, *Codes for the representation of names of countries and their subdivisions — Part 1: Country codes*

ISO 15836, *Information and documentation — The Dublin Core metadata element set*

ISO/IEC 21000-3, *Information technology — Multimedia framework (MPEG-21) — Part 3: Digital Item Identification*

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

ISO/IEC 21000-21, *Information technology — Multimedia framework (MPEG-21) — Part 21: Media Contract Ontology*

XML Encryption Syntax and Processing Version 1.1, W3C Working Draft 16 March 2010,, http://www.w3.org/TR/xmlenc-core1/

IETF RFC 2141, *Uniform Resource Name (URN) Syntax*, May 1997, http://www.ietf.org/rfc/rfc2141.txt

IETF RFC 2396, *Uniform Resource Identifiers (URI)*: Generic Syntax, Internet Standards Track Specification, August 1998, http://www.ietf.org/rfc/rfc2396.txt

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

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

3.1.1
MPEG-21 Contract
representation of agreements formed for the transaction of MPEG-21 Digital Items or services related to the MPEG-21 Framework

3.1.2
MPEG-21 Service
system supplying utility in the Framework of MPEG-21

3.2 Abbreviated terms

For the purposes of this document, the following abbreviated terms apply.

CEL:	Contract Expression Language
IPRE:	Intellectual Property Rights Exploitation
MCO:	Media Contract Ontology
MPEG:	Moving Picture Experts Group
MPEG-21:	ISO/IEC 21000
MPEG-7:	ISO/IEC 15938
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

4.1.1 XML Representation

The syntax of each XML element in the Contract Expression Language is specified using the constructs provided by XML Schema [4]. 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:2012
cel_core.xsd" xmlns:dc="http://purl.org/dc/elements/1.1/" [...]
```

This part of ISO/IEC 21000 also makes use of diagrams to express portions of XML Schema, as shown in Figure 1.

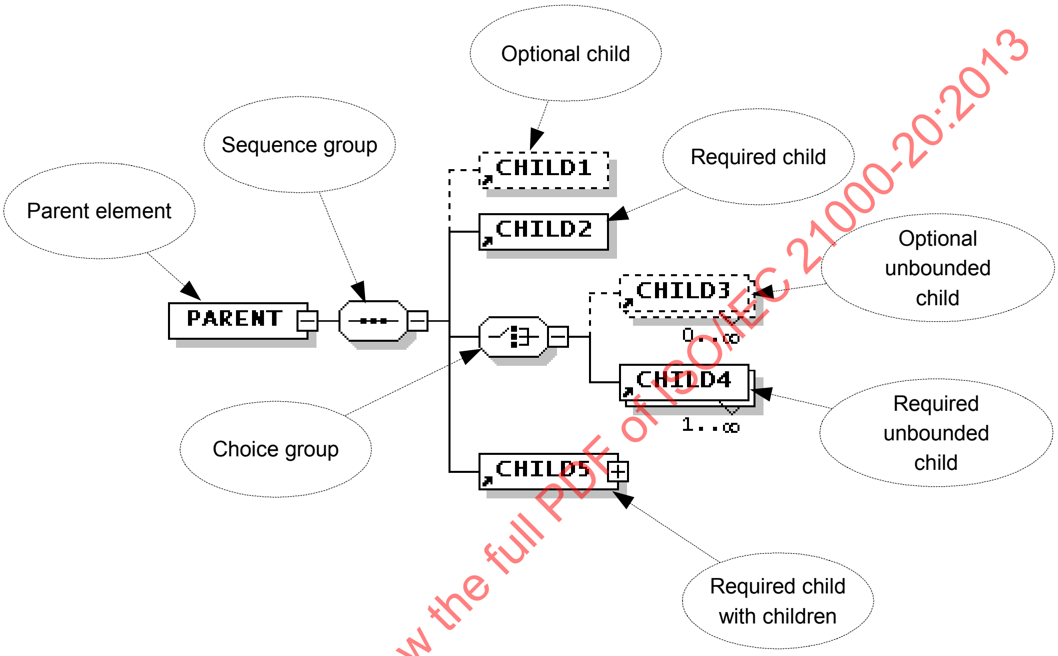


Figure 1 — Sample XML Schema diagram

This schema diagram 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).

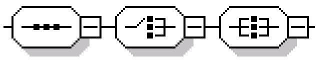


Figure 2 — Compositors used in XML Schema diagrams

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

4.2 Namespace prefix conventions

The namespace for CEL core XML Schema is:

```
urn:mpeg:mpeg21:cel:core:2012
```

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

```
urn:mpeg:mpeg21:cel:ipre:2012
```

4.2.1 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:2012	Here
cel-ipre	urn:mpeg:mpeg21:cel:ipre:2012	Here
dc	http://purl.org/dc/elements/1.1/	ISO 15836
xsi	http://www.w3.org/2001/XMLSchema-instance	[4]
dsig	http://www.w3.org/2000/09/xmldsig#	[1]
rel-r	urn:mpeg:mpeg21:2003:01-REL-R-NS	ISO/IEC 21000-5
dii	urn:mpeg:mpeg21:2002:01-DII-NS	ISO/IEC 21000-3
xenc	http://www.w3.org/2001/04/xmlenc#	W3C XML Encryption Syntax and Processing
xsd	http://www.w3.org/2001/XMLSchema	W3C XML Schema

5 Relationship to other ISO/IEC 21000 parts

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 part of ISO/IEC 21000 with the other parts of ISO/IEC 21000 in addressing the representation of the agreements for the aforementioned transactions.

A contract represented following this part of ISO/IEC 21000 may become a part of a Digital Item (whose declaration is given with ISO/IEC 21000-2). If so, it will be declared with the `Type` element of ISO/IEC 21000-3 pointing to the CEL URI.

This part of ISO/IEC 21000 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 [2][5] or RDF [6]. Both parts share the goal of defining a contract document able to provide the information listed in Clause 6.1.

This part of ISO/IEC 21000 makes use of the semantic for Actions, Facts and Services defined in ISO/IEC 21000-21.

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
- 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 [4]. Any representation of a contract document defined by CEL support the possibility to encrypt either the whole contract or part of it.

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

7 Contract Structured Representation

7.1 Introduction

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

```
urn:mpeg:mpeg21:cel:core:2012
```

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:2012
```

MPEG-21 CEL contract documents shall validate against the XML Schema specification 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 3. It only has one attribute, the `contractId` which uniquely identifies the contract.

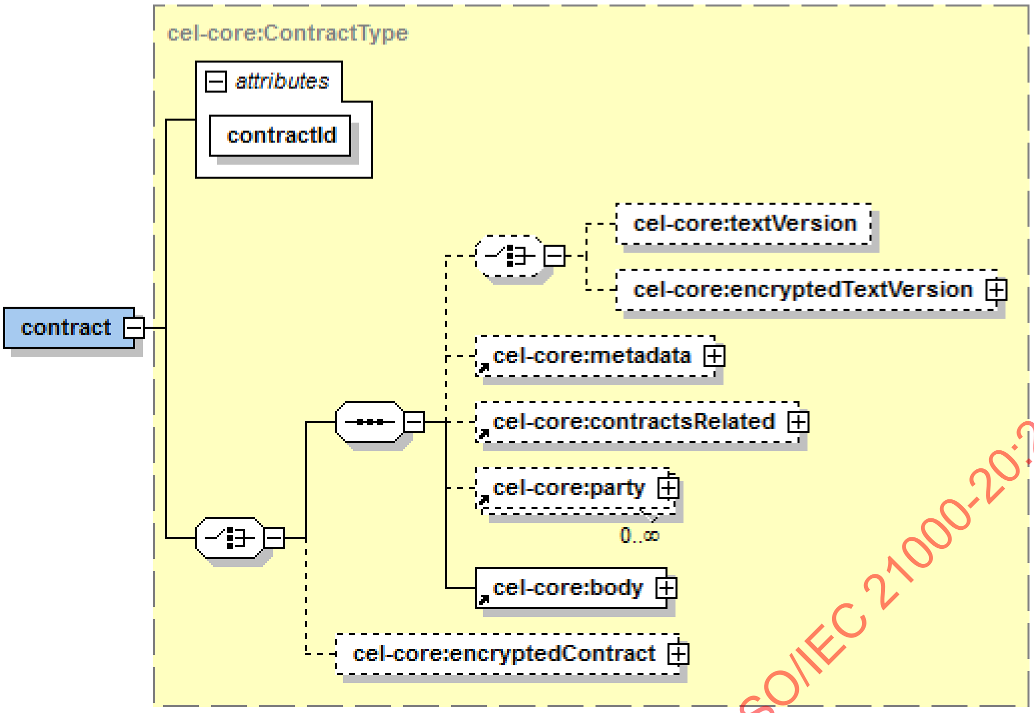


Figure 3 — Root and main elements in the contract

The child elements of the `cel-core:contract` element are:

— 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; 0 means that the contract is a template

— `cel-core:body`, exactly one body is mandatory, with the contract clauses

— `cel-core:EncryptedContract`, a full version of a contract encrypted

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:2012"
   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:2012"
   elementFormDefault="qualified"
   attributeFormDefault="unqualified">
   <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"/>
```

7.2.1.1 Example

A sample contract may reference the XML Schema this way.

```
<?xml version="1.0" encoding="UTF-8"?>
<cel-core:contract xmlns:celxml="urn:mpeg:mpeg21:cel:core:2012"
   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:2012 cel-core.xsd>
```

7.2.2 Metadata

Metadata can be added under the `metadata` element, giving information about the contract itself. Their optional elements can be seen in Figure 4. A DublinCore placeholder is provided (element `cel-core:simpledc` of type `dc:elementContainer`).

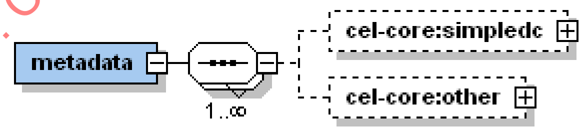


Figure 4 — Contract metadata elements

7.2.3 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 5. The validity of the related contract can be affected according to relationship type, which must 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.

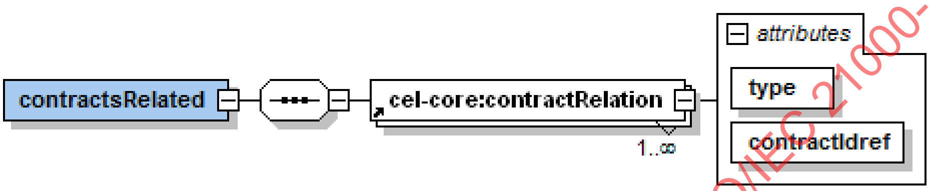


Figure 5 — contractsRelated element

7.2.4 The parties

The contract contains the parties in the contract, as described in Figure 6.

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

A contract template has zero parties, while a contract offer has one party.

Each party can be either a person (element cel-core:person) or an organization (element cel-core: organization). In the latter case, for the party to be binding it is necessary that a signatory person is given (element cel-core:signatory) together with his/her job title (element cel-core:jobTitle) within the organization.

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

— 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 description of the party

— the cel-core:details element, to provide further detailed information.

The party element might be further enriched with the address (element cel-core:address).

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 6.

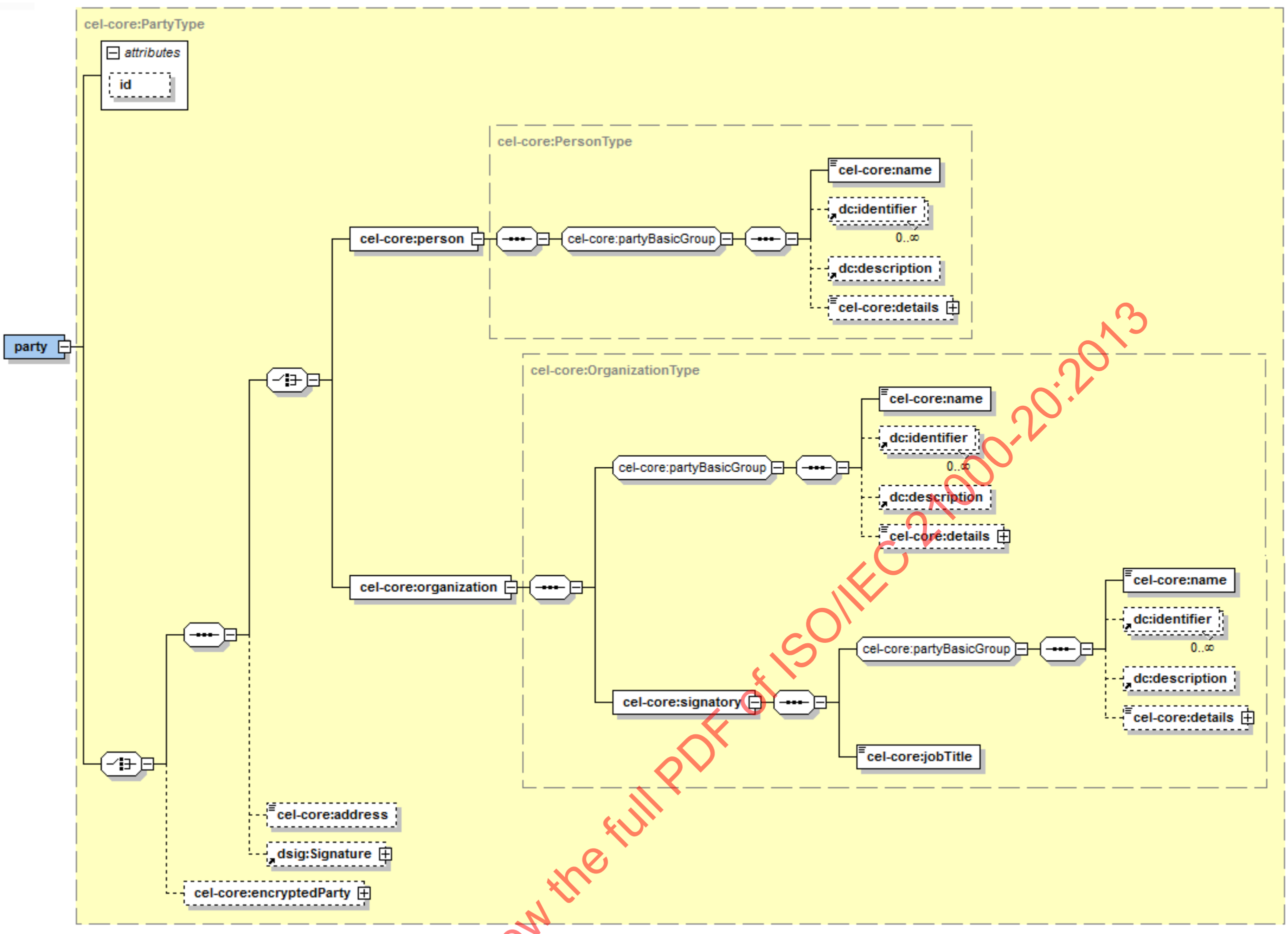


Figure 6 — Contract party element

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

```
<complexType name="PartyType">
   <sequence>
      <choice>
         <element name="person" type="cel-core:PersonType"/>
         <element name="organization" type="cel-core:OrganizationType"/>
      </choice>
      <element name="address" type="string" minOccurs="0"/>
      <element ref="dsig:Signature" minOccurs="0"/>
   </sequence>
   <attribute name="id" type="ID"/>
</complexType>
<group name="partyBasicGroup">
   <sequence>
      <element name="name" type="string"/>
      <element ref="dc:identifier" minOccurs="0" maxOccurs="unbounded"/>
      <element ref="dc:description" minOccurs="0"/>
      <element name="details" type="anyType" minOccurs="0"/>
   </sequence>
</group>
<complexType name="PersonType">
```

```
   <sequence>
       <group ref="cel-core:partyBasicGroup"/>
   </sequence>
</complexType>
<complexType name="OrganizationType">
   <sequence>
       <group ref="cel-core:partyBasicGroup"/>
       <element name="signatory">
          <complexType>
              <sequence>
                 <group ref="cel-core:partyBasicGroup"/>
                 <element name="jobTitle" type="string"/>
              </sequence>
          </complexType>
       </element>
   </sequence>
</complexType>
```

7.2.4.1 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 [3]. Note the use of RDF [6], 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 [7].

```
<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#internet"/>
              </vCard:EMAIL>
            </cel-core:details>
            <cel-core:jobTitle>CEO</cel-core:jobTitle>
         </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>
```

7.2.5 Body element

The `cel-core:body` element contains the contract itself, i.e. the agreed clauses.

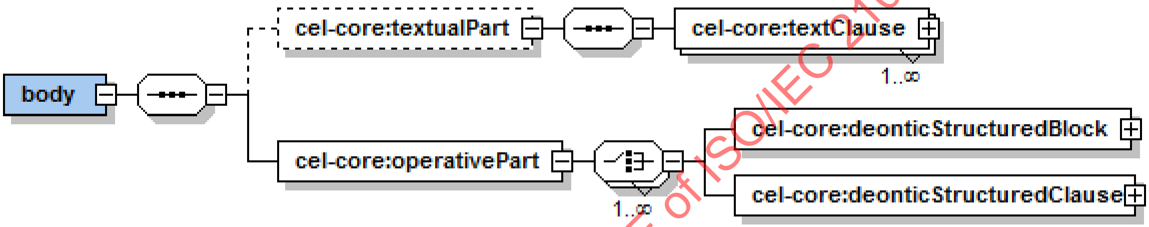


Figure 7 — Structure and placement of body element

The `cel-core:body` element contains the following elements, as can be seen in Figure 7:

— `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

7.2.5.1 Text clauses

Text clauses are represented with elements of type `cel-core:TextClauseType`, which allows a hierarchy of other `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 can be seen in Figure 8.

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. It cannot place the whole contract version in clear text.

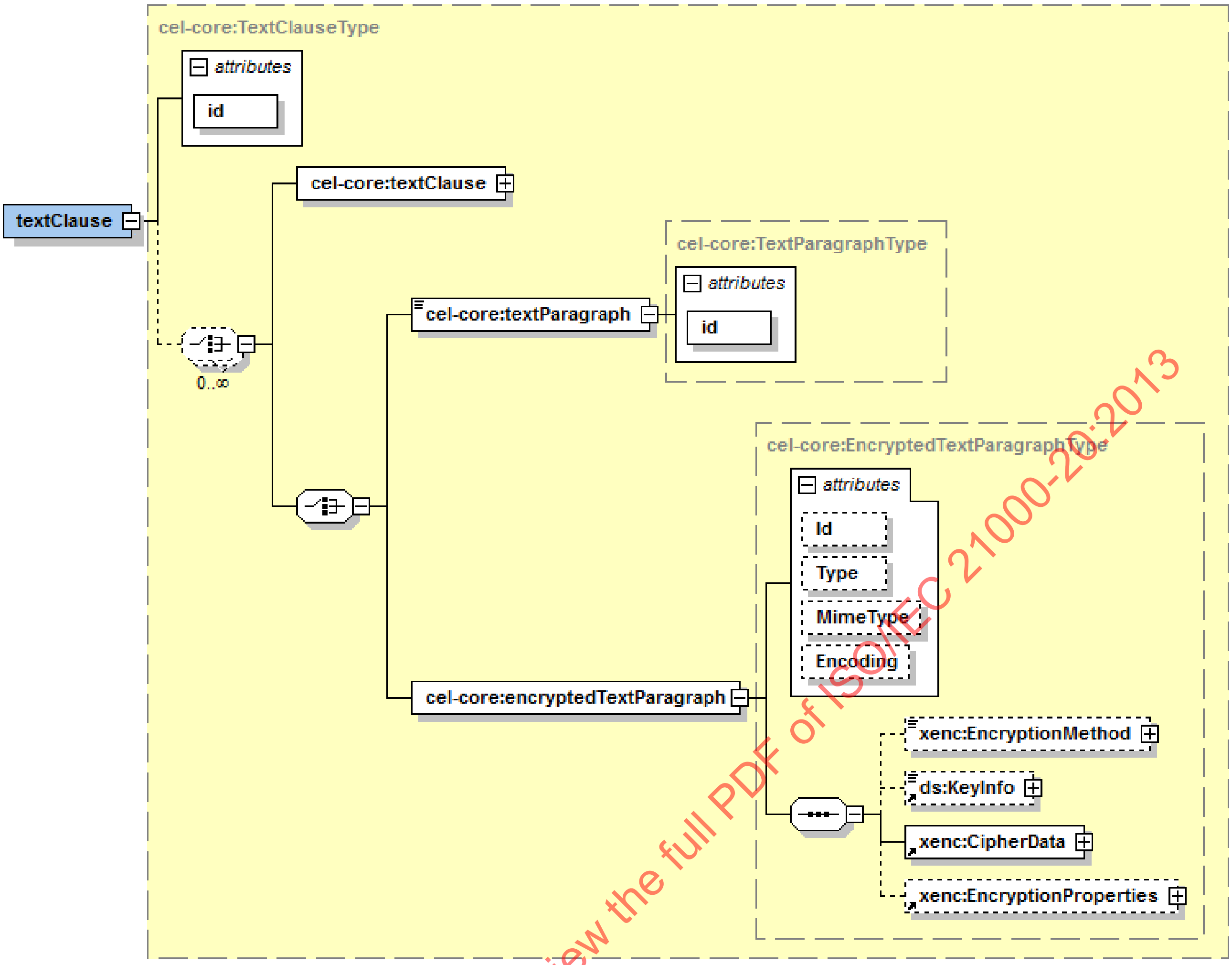


Figure 8 — textClause element

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

```
<complexType name="TextClauseType">
   <choice minOccurs="0" maxOccurs="unbounded">
      <element name="textClause" type="cel-core:TextClauseType"/>
      <choice>
         <element name="textParagraph" type="cel-core:TextParagraphType"/>
         <element name="encryptedTextParagraph" type="cel-
core:EncryptedTextParagraphType"/>
      </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.

7.2.5.1.1 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 Clause 7.2.5.2.1 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 making the Program available to the public, both
free of charge and upon any kind of payments, 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
granted 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 Program 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 Territory 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>
```

7.2.5.2 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 `cel-core:deonticStructuredClause`. Both elements share the same mechanism for referencing elements in the `cel-core:textualPart`, for declaring the type of deontic expression and for allowing the inclusion of encrypted clauses.

7.2.5.2.1 Text references

The reference to elements of the `cel-core:textualPart` is achieved by means of the attribute `idrefs` of type `IDREFS` of the element `cel-core:deonticStructuredClause` or `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.

7.2.5.2.1.1 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 clause 7.2.5.1. The details of the elements `cel-core:deonticStructuredClause`, marked with the character ‘+’, are not shown.

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

7.2.5.2.2 Deontic clause type

MPEG-21 contracts contain agreements between parties, which are represented by deontic concepts of permission, prohibition and obligation.

In an Structured MPEG-21 contract the type of deontic expression of a `cel-core:deonticStructuredClause` or `cel-core:deonticStructuredBlock` is given by the attribute `cel-core:deonticType` of type `cel-core:DeonticType` defined as follows:

```
   <simpleType name="DeonticType">
      <restriction base="string">
         <enumeration value="Permission"/>
         <enumeration value="Obligation"/>
         <enumeration value="Prohibition"/>
         <enumeration value="Statement"/>
      </restriction>
   </simpleType>
```

7.2.5.2.3 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>
```

7.3 CEL Deontic Structure

7.3.1 Introduction

XML 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, child element of the `cel-core:body`.

The `cel-core:operativePart` element is made up of `cel-core:deonticStructuredBlock` and `cel-core:deonticStructuredClause` elements, as shown in Figure 9. The former one enables to group contract clauses which are related in some way, while the latter enables the representation of a clause in a contract.

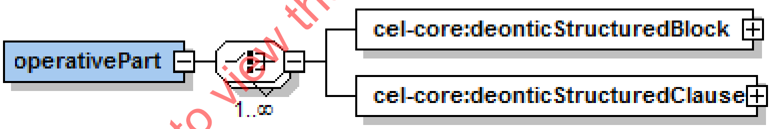

operativePart
1..∞
cel-core:deonticStructuredBlock
cel-core:deonticStructuredClause

Figure 9 — operativePartGroup element

7.3.2 Deontic Structured Block

The `cel-core:deonticStructuredBlock` is defined to group related deontic clauses. It is made up of `cel-core:deonticStructuredBlock`, `cel-core:deonticStructuredClause` and `cel-core:encryptedBlock` elements. Figure 10 shows its structure.

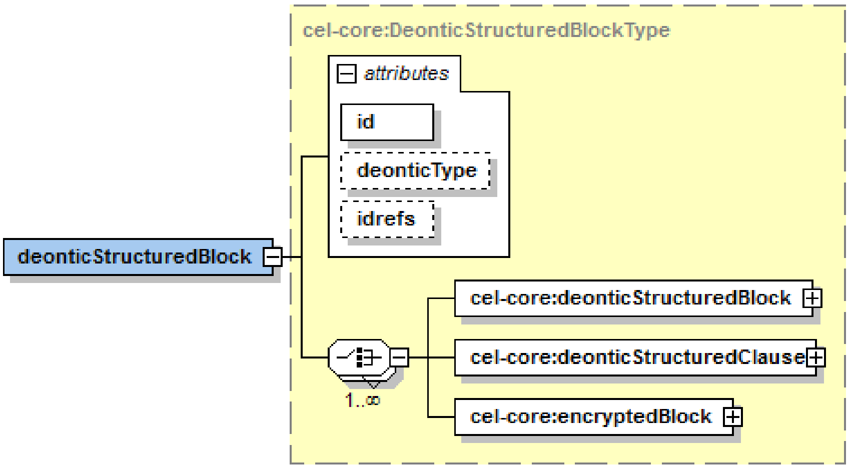

cel-core:DeonticStructuredBlockType
attributes
id
deonticType
idrefs
deonticStructuredBlock
cel-core:deonticStructuredBlock
cel-core:deonticStructuredClause
cel-core:encryptedBlock
1..∞

Figure 10 — 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`

— `deonticType` – to optionally specify the deontic type of the Block. If present, it shall take one of the following values: Permission, Obligation, Prohibition or Statement. It has type `cel-core:DeonticType` (see Annex A)

— `idRefs` – to optionally reference to textual parts of the contract as defined in clause 7.2.5.2.1.

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 Section 7.2.5.2.3).

7.3.3 Deontic Structured Clause

Operative parts of a contract are represented by the `cel-core:deonticStructuredClause element`. A deontic clause represents one permission, obligation or prohibition of a media contract.

The `cel-core:deonticStructuredClause element` consists of the following elements, as depicted in Figure 11:

— `cel-core:metadata` – XML deontic clause metadata

— `cel-core:context` – context for the XML deontic clause

— `cel-core:pre-condition` – 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:post-condition` – post-conditions of the deontic structured clause

- `cel-core:issuer` – party that issues the permission, prohibition, obligation or statement specified in the deontic clause

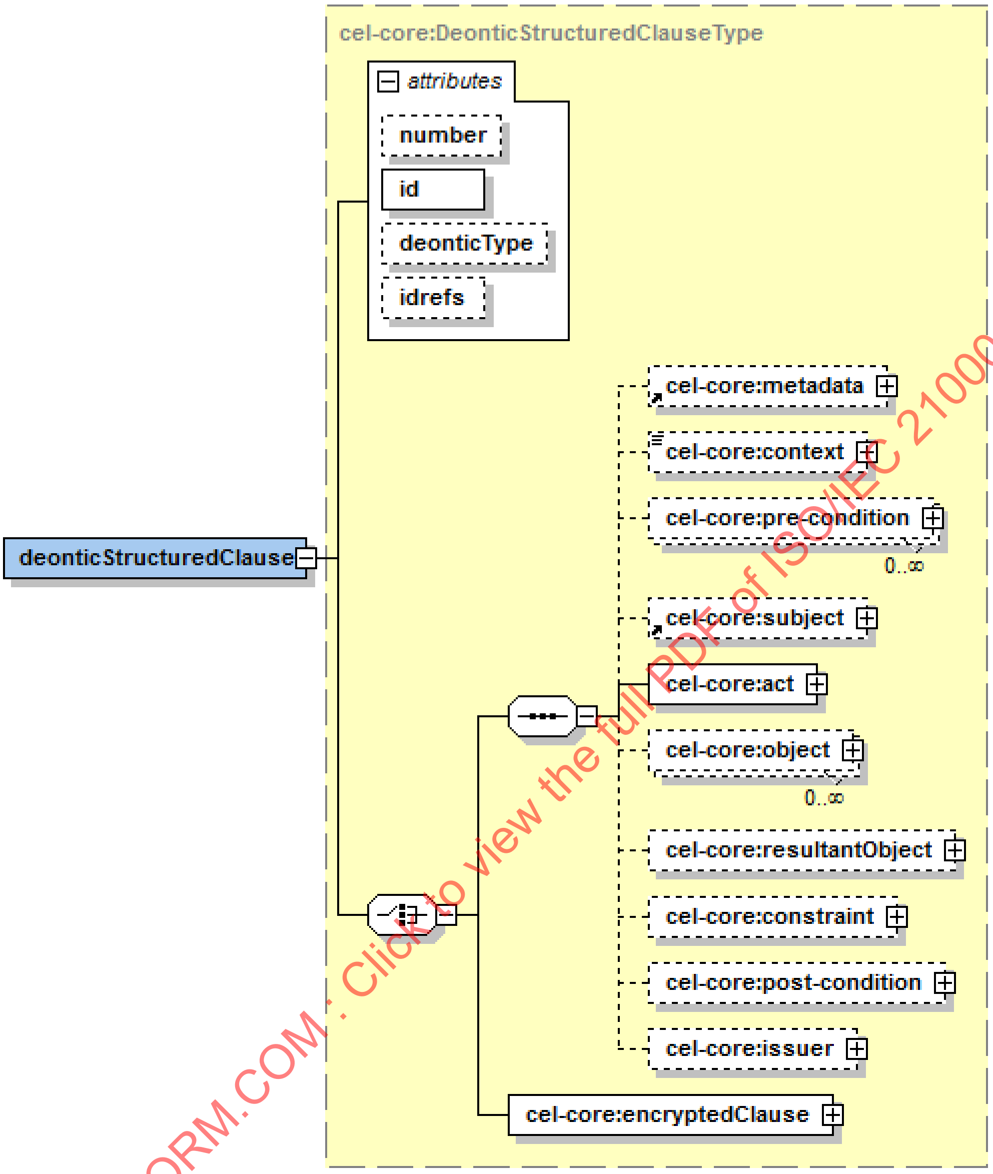


Figure 11 — deonticStructuredClause element

The `cel-core:deonticStructuredClause` element has the 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`

- `deonticType` – to optionally specify the deontic type of the clause. If present, it shall take one of the following values: Permission, Obligation, Prohibition or Statement. It has type `cel-core:DeonticType` (see Annex A)

- `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.

7.3.3.1 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 clause 7.2.2, its structure is shown in Figure 12.

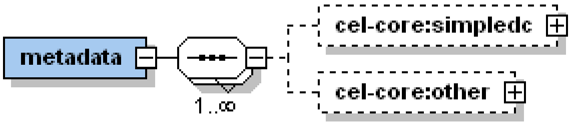


Figure 12 — metadata element

7.3.3.2 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 13.

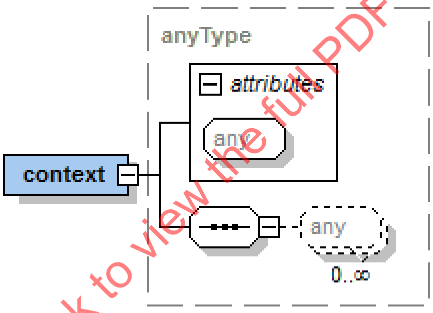


Figure 13 — context element

7.3.3.3 Pre-condition

Pre-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, pre-conditions can be stated in deontic clauses by means of the cel-core:pre-condition element, referring to an existing cel-core:deonticStructuredBlock or cel-core:deonticStructuredClause by means of the idref attribute, as depicted in Figure 14.

The cel-core:pre-condition also has the attributes:

- actionStatus to optionally indicate the status of the action in the deontic expression to which the cel-core:pre-condition 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:pre-condition after which the permission, obligation or prohibition granted by the deontic expression to which the cel-core:pre-condition pertains has to be considered valid

— `hasValidity` to optionally determine the time of validity of the permission, obligation or prohibition granted by the deontic expression to which the `cel-core:pre-condition` pertains

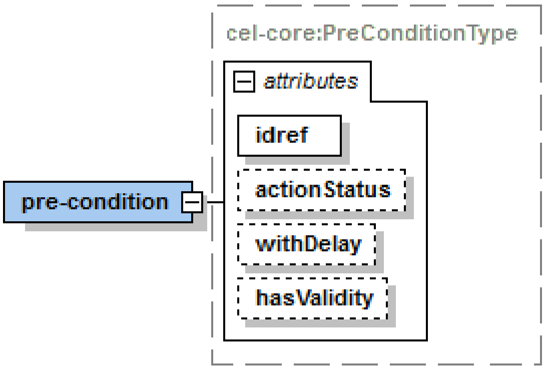


Figure 14 — pre-condition element

7.3.3.4 Subject

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 `cel-core:partyRef` attribute, as shown in Figure 15.

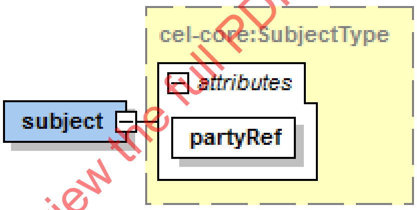


Figure 15 — subject element

7.3.3.4.1 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:deonticStructuredClause id="p001" deonticType="Permission">
    <cel-core:subject partyRef="user002"/>
    <cel-core:act> ... </cel-core:act>
    <cel-core:object> ... </cel-core:object>
    ...
   </cel-core:deonticStructuredClause>
  </cel-core:operativePart>
 </cel-core:body>
</cel-core:contract>
```

7.3.3.5 Act

The `cel-core:act` element specifies the right that will be granted to the `cel-core:subject` if the constraints, pre-conditions and post-conditions (if present) are met.

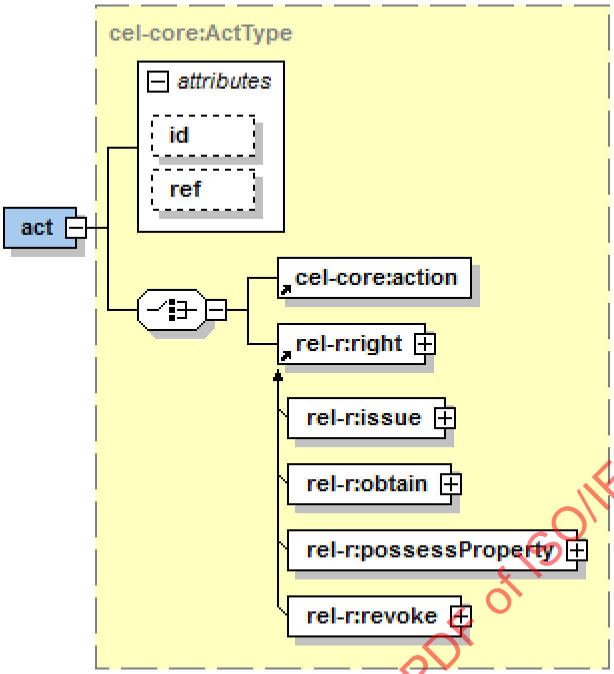


Figure 16 — act element

The acts in a media contract include those defined in the CEL Extension for Exploitation of Intellectual Property Rights, see clause 7.4.1, 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 16.

7.3.3.5.1 Example

```
<cel-core:contract>
 ...
 <cel-core:body>
  <cel-core:operativePart>
   <cel-core:deonticStructuredClause id="p001" deonticType="Permission">
    <cel-core:subject> ... </cel-core:subject>
    <cel-core:act>
     <cel-ipre:makeAdaptation/>
    </cel-core:act>
    <cel-core:object> ... </cel-core:object>
    ...
   </cel-core:deonticStructuredClause>
  </cel-core:operativePart>
 </cel-core:body>
</cel-core:contract>
```

7.3.3.6 Object

The `cel-core:object` element defines the resource against which the right will apply.

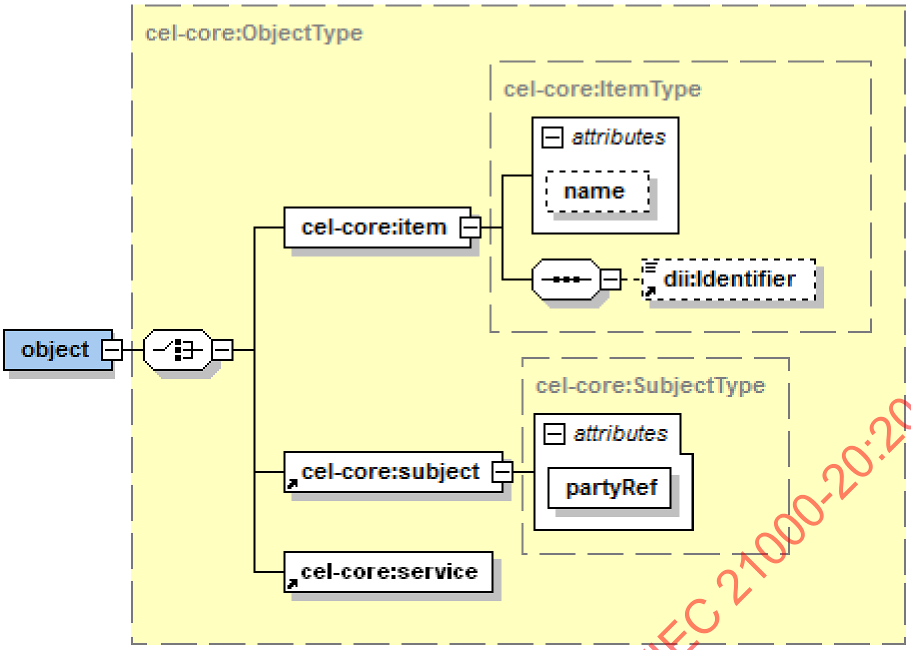


Figure 17 — object element

Objects are defined in deontic clauses under the `cel-core:object` element, as shown in Figure 17. Three different types of objects are supported:

- a digital resource identified by the `dii:Identifier` element of the `cel-core:item` element. It can be optionally named by means of the `cel-core:name` attribute
- one of the parties of the contract referenced by the `cel-core:partyRef` attribute of the `cel-core:subject` element
- an MPEG service, like authentication, identification, encryption, processing, search or others (see clause 7.4.3)

7.3.3.6.1 Example

```
<cel-core:contract>
 ...
 <cel-core:body>
  <cel-core:operativePart>
   <cel-core:deonticStructuredClause id="p001" deonticType="Permission">
    <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:deonticStructuredClause>
  </cel-core:operativePart>
 </cel-core:body>
</cel-core:contract>
```

7.3.3.7 Resultant Object

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 cel-core:name attribute, as shown in Figure 18.

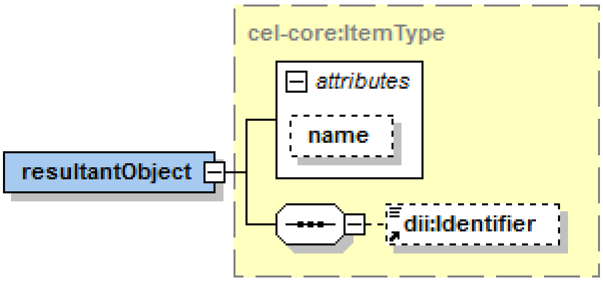


Figure 18 — resultantObject element

7.3.3.7.1 Example

```
<cel-core:contract>
 ...
 <cel-core:body>
  <cel-core:operativePart>
   <cel-core:deonticStructuredClause id="p001" deonticType="Permission">
    <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:deonticStructuredClause>
  </cel-core:operativePart>
 </cel-core:body>
</cel-core:contract>
```

7.3.3.8 Constraint

Conditions, restrictions and constraints can be specified under the cel-core:constraint element, as depicted in Figure 19.

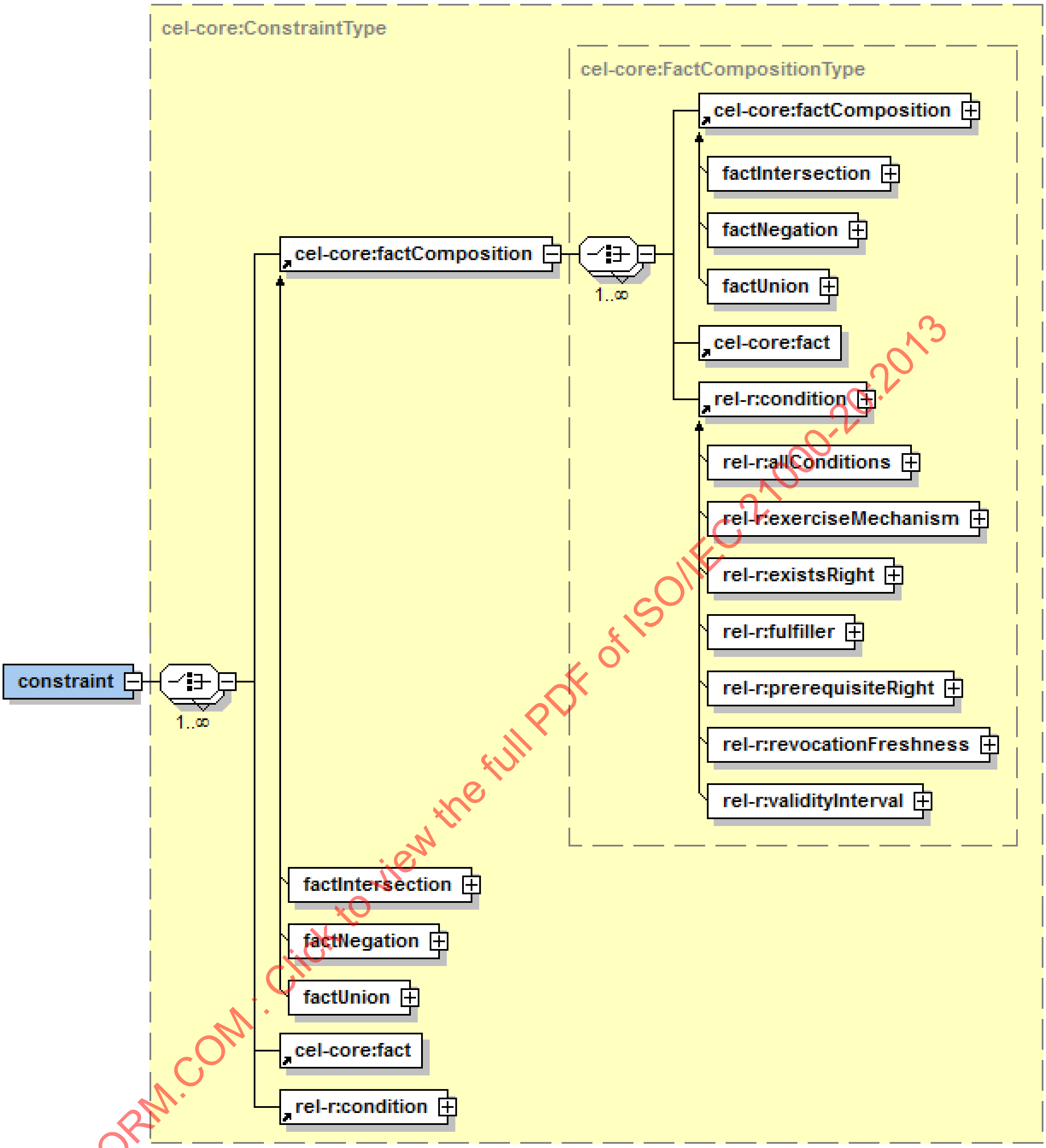


Figure 19 — constraint element

Simple conditions, restrictions and constraints of a deontic clause include those defined in the CEL Extension for Exploitation of Intellectual Property Rights, see clause 7.4.2, under the `cel-core:constraint`, `cel-core:factComposition`, `cel-core:factIntersection`, `cel-core:factNegation` or `cel-core:factUnion` elements replacing the `cel-core:fact` element or in ISO/IEC 21000-5 replacing the `rel-r:condition` element.

Complex conditions are represented by means of the `cel-core:factComposition`, `cel-core:factIntersection`, `cel-core:factNegation` and `cel-core:factUnion` elements.

The `cel-core:factComposition` element provides the means for grouping logical constructs consisting on multiple complex and/or simple facts. Its truth depends on one or more facts according to the logical constructs defined by its child 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 the only fact which form 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.

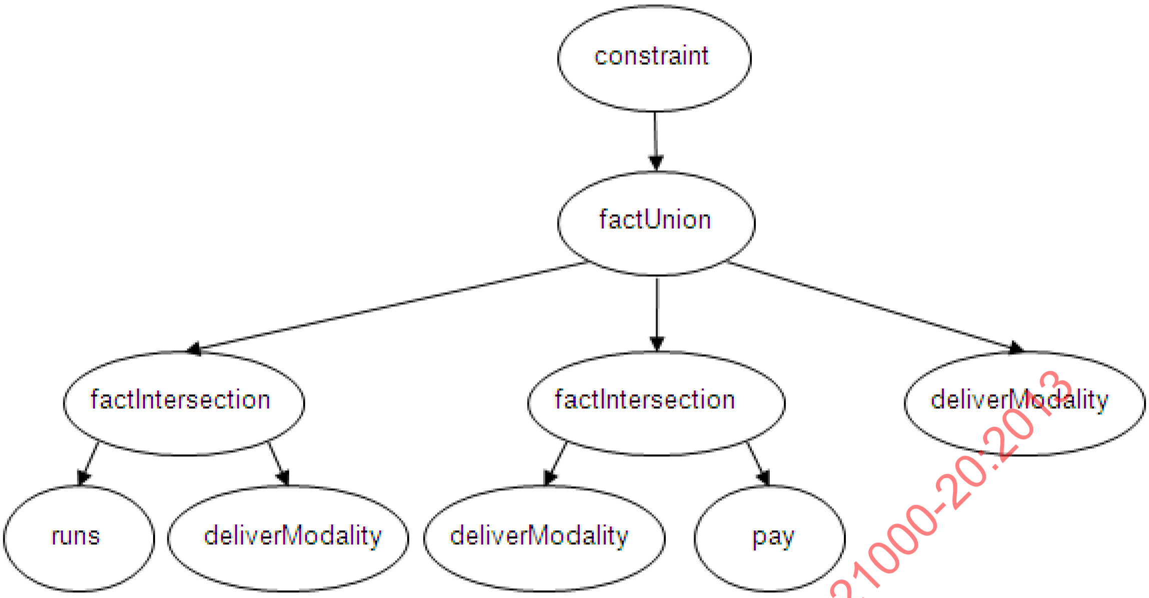
7.3.3.8.1 Simple constraint example

Next box shows an example of a temporal Interval constraint.

```
<cel-core:contract>
 ...
 <cel-core:body>
  <cel-core:operativePart>
   <cel-core:deonticStructuredClause id="p001" deonticType="Permission">
    <cel-core:subject> ... </cel-core:subject>
    <cel-core:act> ... </cel-core:act>
    <cel-core:object> ... </cel-core:object>
    <cel-core:constraint>
     <cel-ipre:temporalInterval>
      <cel-ipre:afterDate>2011-07-22T00:00:00</cel-ipre:afterDate>
      <cel-ipre:beforeDate>2001-11-30T23:59:59</cel-ipre:beforeDate>
     </cel-ipre:temporalInterval>
    </cel-core:constraint>
   </cel-core:deonticStructuredClause>
  </cel-core:operativePart>
 </cel-core:body>
</cel-core:contract>
```

7.3.3.8.2 Complex constraint example

Next 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 pay; and the third, the deliveryModality constraint.



```
<cel:contract>
 ...
 <cel-core:body>
  <cel-core:operativePart>
   <cel-core:deonticStructuredClause id="p001" deonticType="Permission">
    <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 number="10"/>
       < cel-ipre:deliveryModality mod="Linear"/>
      </cel-core:factIntersection>
      <cel-core:factIntersection>
       <cel-ipre:deliveryModality mod="Linear"/>
       <cel-ipre:pay/>
      </cel-core:factIntersection>
      <cel-ipre:deliveryModality mod="NonLinear"/>
     </cel-core:factUnion>
    </cel-core:constraint>
   </cel-core:deonticStructuredClause>
  </cel-core:operativePart>
 </cel-core:body>
</cel-core:contract>
```

7.3.3.9 Post-condition

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:post-condition` element, referring to an existing `cel-core:deonticStructuredBlock` or `cel-core:deonticStructuredClause` by means of the `idref` attribute, as depicted in Figure 20.

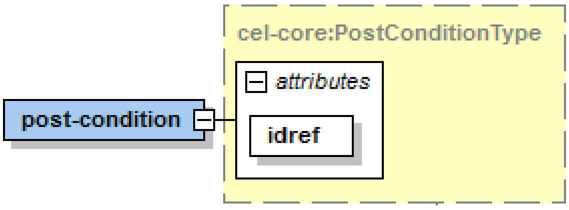


Figure 20 — post-condition element

7.3.3.10 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 21.

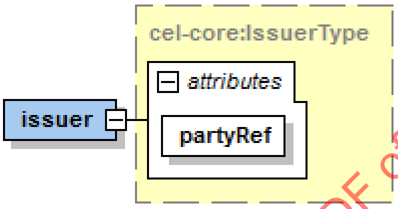


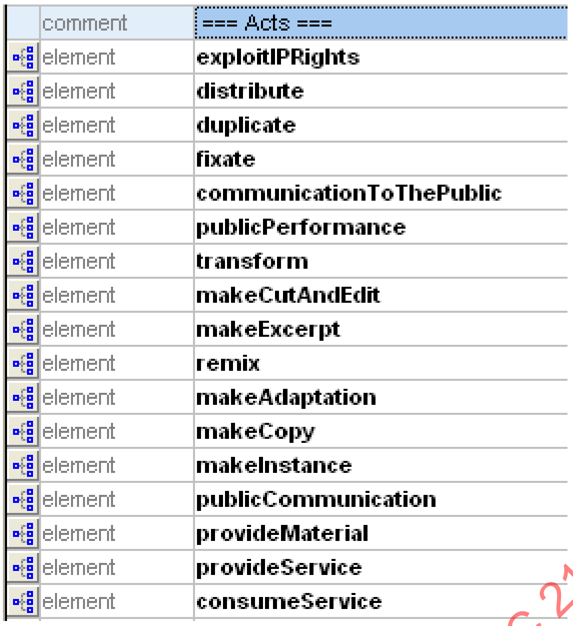
Figure 21 — issuer element

7.4 CEL Extension for Exploitation of Intellectual Property Rights

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:2012`.

7.4.1 Acts

The acts required for modeling rights in audiovisual contracts are depicted in Figure 22. The semantic for acts is provided in Clause 8.2.2 of ISO/IEC 21000-21, as well as their hierarchy. Acts include the IP-rights of an audiovisual contract, as well as acts related with the offering and consumption of services.



comment	=== Acts ===
element	exploitIPRights
element	distribute
element	duplicate
element	fixate
element	communicationToThePublic
element	publicPerformance
element	transform
element	makeCutAndEdit
element	makeExcerpt
element	remix
element	makeAdaptation
element	makeCopy
element	makeInstance
element	publicCommunication
element	provideMaterial
element	provideService
element	consumeService

Figure 22 — Media Contracts CEL extension Acts

7.4.1.1 ProvideMaterial

The `cel-ipre:provideMaterial` element specifies the act of transferring the content to the parties referenced by the `providedTo` 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-ipre:provideMaterial` element is depicted in Figure 23.

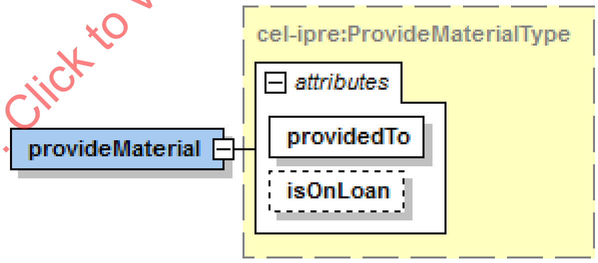
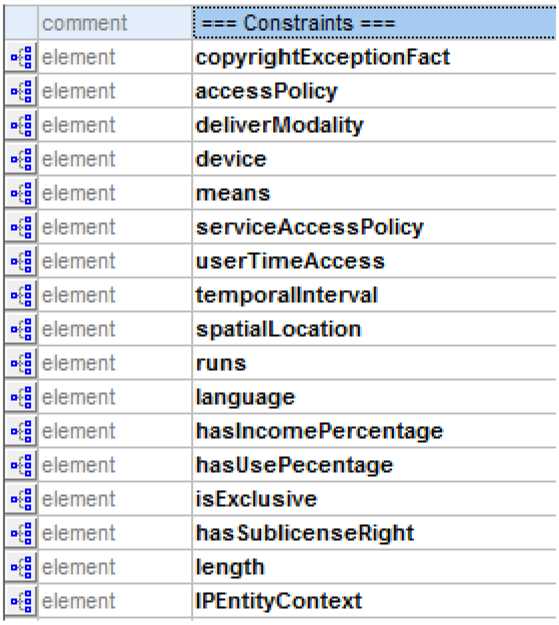


Figure 23 — provideMaterial act

7.4.2 Constraints

The constraints required for modeling conditions in audiovisual contracts are depicted in Figure 24. The semantic for constraints is provided in next subclauses. A hierarchy for the constraints is defined in clause 8.2.2.3 of ISO/IEC 21000-21 which describes the MCO Extension for Exploitation of Intellectual Property Rights.



comment	=== Constraints ===
element	copyrightExceptionFact
element	accessPolicy
element	deliverModality
element	device
element	means
element	serviceAccessPolicy
element	userTimeAccess
element	temporalInterval
element	spatialLocation
element	runs
element	language
element	hasIncomePercentage
element	hasUsePecentage
element	isExclusive
element	hasSublicenseRight
element	length
element	IPEntityContext

Figure 24 — Media Contracts CEL extension Constraints

7.4.2.1 AccessPolicy

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

- `freeOfCharge`: the content is accessible for free or only fees levied by government agencies can be applied
- `pay`: the content is accessible through payment
- `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

```
<complexType name="AccessPolicyType">
 <complexContent>
  <extension base="cel-core:Fact">
   <attribute name="access" type="cel-ipre:AccessType" use="required"/>
  </extension>
 </complexContent>
</complexType>
<simpleType name="AccessType">
 <restriction base="string">
  <enumeration value="freeOfCharge"/>
  <enumeration value="pay"/>
  <enumeration value="subscription"/>
  <enumeration value="payPerView"/>
 </restriction>
</simpleType>
```

7.4.2.2 DeliveryModality

The cel-ipre:deliveryModality constraint restricts the fruition of the communication to the public on who and how has the control and on the time and place by means of the cel-ipre:mod attribute which should take one of the following values:

- linear: the media content is simultaneously delivered to end users by a service provider
- 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
- onDemandBasis: the media content is delivered to an end user as response to an individual request
- 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 a individual request

```
<complexType name="DeliverType">
 <complexContent>
  <extension base="cel-core:Fact">
   <attribute name="mod" type="cel-ipre:ModalityType" use="required"/>
  </extension>
 </complexContent>
</complexType>
<simpleType name="ModalityType">
 <restriction base="string">
  <enumeration value="linear"/>
  <enumeration value="broadcasting"/>
  <enumeration value="webcasting"/>
  <enumeration value="nonLinear"/>
  <enumeration value="onDemandBasis"/>
  <enumeration value="onDemandDownload"/>
  <enumeration value="onDemandStreaming"/>
 </restriction>
</simpleType>
```

7.4.2.3 Device

The cel-ipre:device constraint restricts the type of equipment used for the fruition of the content according to the value given to its attribute cel-ipre:type, which 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
- 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

- televisionSet: the end user rendering device is any equipment with television functionalities, but is not a mobile broadcast device

```
<complexType name="DeviceType">
 <complexContent>
  <extension base="cel-core:Fact">
   <attribute name="type" type="cel-ipre:DeviceT" use="required"/>
  </extension>
 </complexContent>
</complexType>
<simpleType name="DeviceT">
 <restriction base="string">
  <enumeration value="computer"/>
  <enumeration value="mobileDevice"/>
  <enumeration value="mobileBroadcastDevice"/>
  <enumeration value="mobileTelecommunicationDevice"/>
  <enumeration value="robotDevice"/>
  <enumeration value="storageDevice"/>
  <enumeration value="televisionDevice"/>
  <enumeration value="televisionSet"/>
 </restriction>
</simpleType>
```

7.4.2.4 Means

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

- broadcastTechnology: the technology used for delivering media content is point-to-multipoint

- 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)

- download: the protocol used for retrieving media content from a network is the HTTP protocol (or equivalent)

- upload: the protocol used for storing media content in a network is the HTTP protocol (or equivalent)

- mobileTechnology: the technology used for delivering media content is suitable for rendering in a mobile device

- mobileTelecommunicationTechnology: the technology used for delivering media content is point-to-point and the content will be rendered on portable devices

- radio: the technology used for delivering media content is Radio

- videogram: the technology used for delivering media content support the creation of a media content container in any format (e.g. videocassettes, DVD, DVD-RIM, internet access ready DVD, CDi, CD, CD-ROM, UMD, VCD)

```
<complexType name="MeansType">
 <complexContent>
  <extension base="cel-core:Fact">
   <attribute name="tec" type="cel-ipre:TecType" use="required"/>
  </extension>
 </complexContent>
</complexType>
<simpleType name="TecType">
 <restriction base="string">
  <enumeration value="broadcastTechnology"/>
  <enumeration value="cable"/>
  <enumeration value="IPNetwork"/>
  <enumeration value="mobileBroadcastTechnology"/>
  <enumeration value="satellite"/>
  <enumeration value="terrestrial"/>
  <enumeration value="internet"/>
  <enumeration value="download"/>
  <enumeration value="upload"/>
  <enumeration value="mobileTechnology"/>
  <enumeration value="mobileBroadcastTechnology"/>
  <enumeration value="mobileTelecommunicationTechnology"/>
  <enumeration value="radio"/>
  <enumeration value="videogram"/>
 </restriction>
</simpleType>
```

7.4.2.5 ServiceAccessPolicy

The cel-ipre:serviceAccessPolicy constraint restricts the access to services according to the value given to its attribute cel-ipre:mod 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

```
<complexType name="ServiceAccessPolicyType">
 <complexContent>
  <extension base="cel-core:Fact">
   <attribute name="mod" type="cel-ipre:ModType" use="required"/>
  </extension>
 </complexContent>
</complexType>
<simpleType name="ModType">
 <restriction base="string">
  <enumeration value="open"/>
  <enumeration value="restricted"/>
 </restriction>
</simpleType>
```

7.4.2.6 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 `hasValidity` attribute that determines how long the user can access after the first access. Otherwise, it has the `cel-ipre:unlimited` child element.

Figure 25 shows the `cel-ipre:userTimeAccess` element structure.

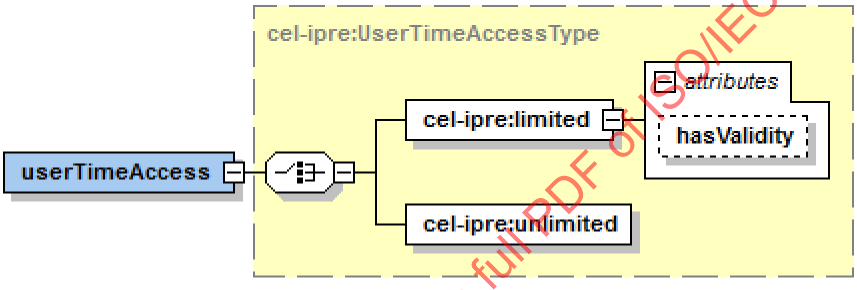


Figure 25 — userTimeAccess constraint

7.4.2.7 Temporal Interval

The `cel-ipre:temporalInterval` constraint restricts the time period of execution of the action specified in the deontic clause. It consists of two child elements, as shown in Figure 26:

- `cel-ipre:afterDate`: date after which the action can be executed
- `cel-ipre:beforeDate`: date before which the action can be executed

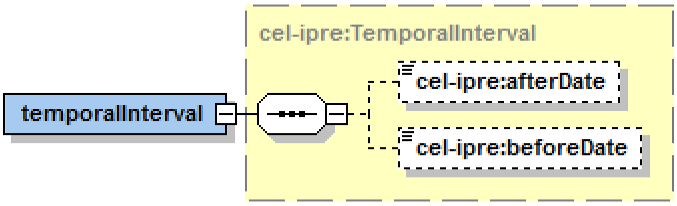


Figure 26 — temporalInterval constraint

7.4.2.8 Spatial Location

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

Figure 27 shows the `cel-ipre:spatialLocation` element structure.

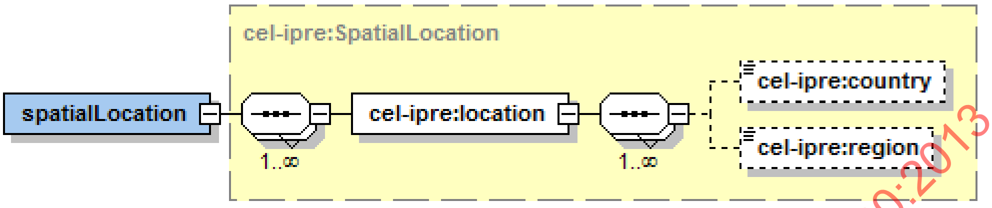


Figure 27 — spatialLocation constraint

7.4.2.9 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 28:

- `cel-ipre:number`: maximum number of allowed runs
- `cel-ipre:hasNumerOfRepetitions`: specifies the number of repetitions (within a validity window) to be considered a single run. It is optional
- `cel-ipre:hasValidity`: period during which an unlimited or specified number of repetitions have to be considered as a single run. It is optional

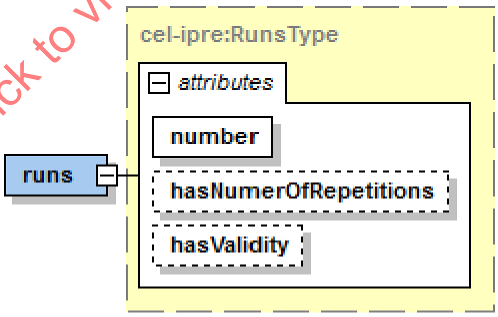


Figure 28 — runs constraint

7.4.2.10 Language

The `cel-ipre:language` constraint states the languages permitted for public communication. It has the `cel-ipre:attribute`, as shown in Figure 29, in which the languages are specified, for example by means of an ISO code.

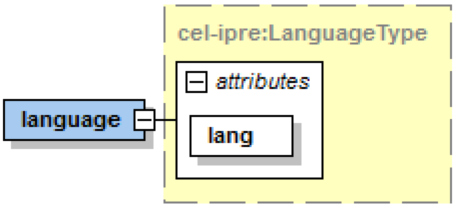


Figure 29 — language constraint

7.4.2.11 HasIncomePercentage

The `cel-ipre:hasIncomePercentage` constraint determines the income of the exploitation of the permission by means of its attribute `cel-ipre:perc`. Figure 30 shows the structure of the `cel-ipre:hasIncomePercentage` element.

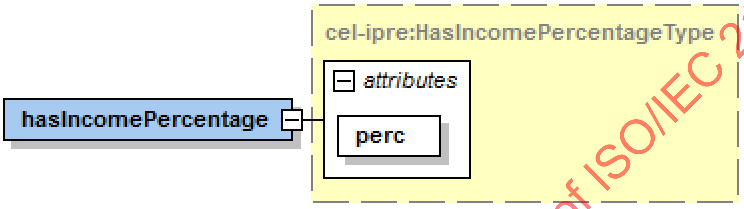


Figure 30 — hasIncomePercentage constraint

7.4.2.12 HasUsePercentage

The `cel-ipre:hasUsePercentage` constraint is used when the permission is shared with other actors by means of its attribute `cel-ipre:perc`. Figure 31 shows the structure of the `cel-ipre:hasUsePercentage` element.

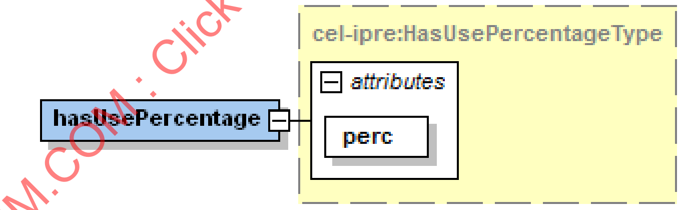


Figure 31 — hasUsePercentage constraint

7.4.2.13 IsExclusive

The `cel-ipre:isExclusive` constraint determines if this exploitation might be granted by the issuer to multiple licensees by means of its attribute `cel-ipre:value`. Figure 32 shows the structure of the `cel-ipre:isExclusive` element.

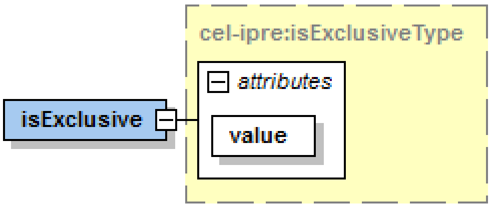


Figure 32 — isExclusive constraint

7.4.2.14 HasSublicenseRight

The cel-ipre:hasSublicenseRight constraint restricts if the granted action can be sublicensed by means of its attribute cel-ipre:value, which has a boolean value. Figure 33 shows the structure of the cel-ipre:hasSublicenseRight element.

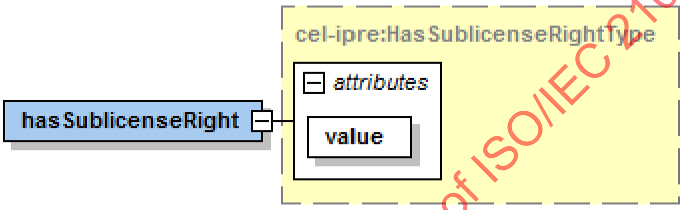


Figure 33 — hasSublicenseRight constraint

7.4.2.15 Length

The cel-ipre:length constraint restricts the length in terms of duration (canonical play) by means of its attribute cel-ipre:hasMaxLength which specifies the maximum length of an object resultant from an action. The cel-ipre:hasMaxLength attribute has type xsd:duration. Figure 34 shows the structure of the cel-ipre:length element.

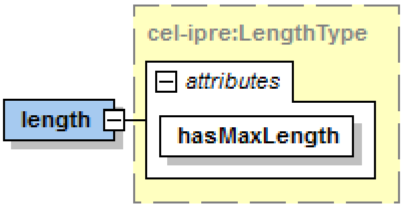


Figure 34 — length constraint

7.4.2.16 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 of its child element cel-ipre:partOf. Figure 35 shows the structure of the cel-ipre:IPEntityContext element.

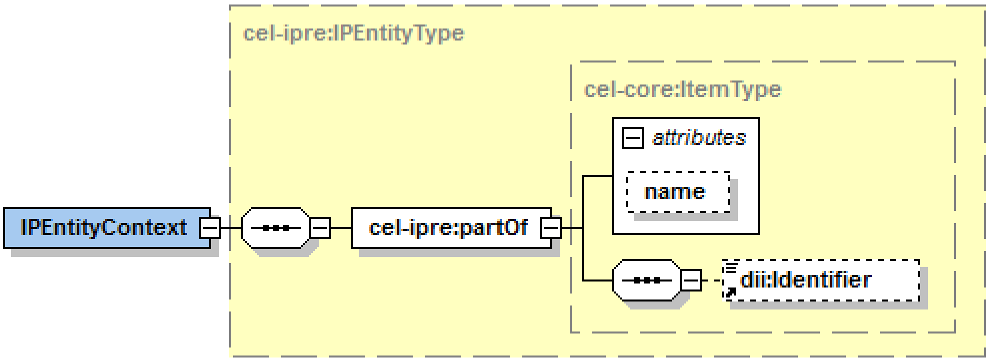
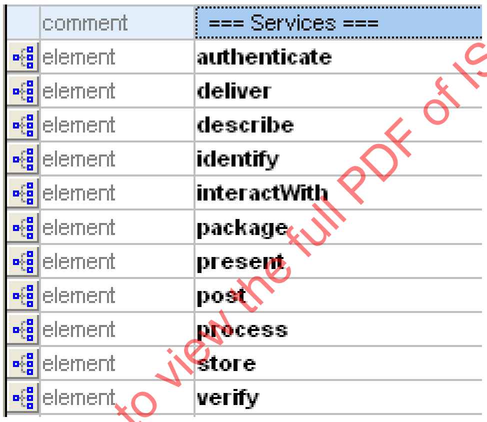


Figure 35 — IPEntityContext constraint

7.4.3 Services

The services required for modeling rights in audiovisual contracts are depicted in Figure 36. The semantic for services is provided in Clause 8.1.2 of ISO/IEC 21000-21.



comment	=== Services ===
element	authenticate
element	deliver
element	describe
element	identify
element	interactWith
element	package
element	present
element	post
element	process
element	store
element	verify

Figure 36 — CEL services

7.5 Examples

This clause provides the XML representation for four sample contracts, which include a simple permission, simple prohibition, simple obligation, and complex permission.

7.5.1 Simple Permission

This clause 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:deonticStructuredClause` element, giving the `Permission` value to its `deonticType` attribute.

```
<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:deonticStructuredClause id="p001" deonticType="Permission">
    <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:deonticStructuredClause>
  </cel-core:operativePart>
 </cel-core:body>
</cel-core:contract>
```

7.5.2 Simple Prohibition

This clause 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:deonticStructuredClause` element, giving the `Prohibition` value to its `deonticType` attribute.

```
<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:deonticStructuredClause id="p002" deonticType="Prohibition">
    <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:deonticStructuredClause>
  </cel-core:operativePart>
 </cel-core:body>
</cel-core:contract>
```

7.5.3 Simple Obligation

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

An obligation is expressed by means of the `cel-core:deonticStructuredClause` element, giving the `Obligation` value to its `deonticType` attribute.

```
<cel-core:contract>
 <cel-core:metadata>
  <cel-core:simpledc>
   <dc:title>Example Obligation o001</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:deonticStructuredClause id="o001" deonticType="Obligation">
    <cel-core:subject partyRef="user002"/>
    <cel-core:act>
     <cel-ipre:audiovisualDistribute/>
    </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:temporalInterval>
      <cel-ipre:afterDate>2011-07-22T00:00:00</cel-ipre:afterDate>
      <cel-ipre:beforeDate>2001-11-30T23:59:59</cel-ipre:beforeDate>
     </cel-ipre:temporalInterval>
    </cel-core:constraint>
    <cel-core:issuer partyRef="user001"/>
   </cel-core:deonticStructuredClause>
  </cel-core:operativePart>
 </cel-core:body>
</cel-core:contract>
```

7.5.4 Complex permission

This clause provides the XML representation of a more complex permission, where the user `User0002` is allowed to `communicateToThePublic` the `AnimatedSeries1`, but subject to certain constraints (spatial and temporal), in the countries of Italy, San Marino and the Vatican between the 25th September 2009 and 30th September 2013.

```
<cel-core:contract>
 <cel-core:metadata>
  <cel-core:simpledc>
   <dc:title>Example Permission p004</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:deonticStructuredClause id="p004" deonticType="Permission">
    <cel-core:subject partyRef="user002"/>
    <cel-core:act>
     <cel-ipre:communicationToThePublic/>
    </cel-core:act>
    <cel-core:object>
     <cel-core:item>
      <dii:Identifier>AnimatedSeries1</dii:Identifier>
     </cel-core:item>
    </cel-core:object>
    <cel-core:constraint>
     <cel-ipre:temporalInterval>
      <cel-ipre:afterDate>2009-09-25T00:00:00</cel-ipre:afterDate>
      <cel-ipre:beforeDate>2013-09-30T23:59:59</cel-ipre:beforeDate>
     </cel-ipre:temporalInterval>
     <cel-ipre:spatialLocation>
      <cel-ipre:country>iso:IT</cel-ipre:country>
      <cel-ipre:country>iso:SM</cel-ipre:country>
      <cel-ipre:country>iso:VA</cel-ipre:country>
     </cel-ipre:spatialLocation>
    </cel-core:constraint>
    <cel-core:issuer partyRef="user001"/>
   </cel-core:deonticStructuredClause>
  </cel-core:operativePart>
 </cel-core:body>
</cel-core:contract>
```

7.6 CEL Extension mechanism (informative)

ISO/IEC 21000-20 is structured in two main parts, the CEL Core, which provides an extensible structure for digitally representing media contracts, and the CEL Extension for Exploitation of Intellectual Property Rights, which includes the most common acts and constraints in the media field, to be later used in digital media contracts.

It is expected that new extensions will be constructed for other domains or applications. To this end, ISO/IEC 21000-20 provides an extension mechanism detailed in this clause.

An XML extension is, in this context, a set of new XML elements, types, URIs and QNames to be used in conjunction with the CEL Core and optionally the CEL Extension for Exploitation of Intellectual Property Rights and/or the future ones. The new elements and types, mainly acts, services and constraints, will be derived from elements and types existing in the CEL Core, or even in one of its extensions (e..g. Exploitation of Intellectual Property Rights Extension). The new elements will derive from existing elements making use of the extension mechanism provided by the `xsd:substitutionGroup` element.

For example, if a new extension in the e-health domain is created, with namespace `urn:mpeg:mpeg21:cel:ehd:2012`, and with the acts `review` and `archive`, and the constraint `ipAccess`, a new XML Schema would be defined as follows:

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
        xmlns:cel-core="urn:mpeg:mpeg21:cel:core:2012"
        xmlns:cel-ehd="urn:mpeg:mpeg21:cel:ehd:2012"
        targetNamespace="urn:mpeg:mpeg21:cel:ipre:2012"
        elementFormDefault="qualified" attributeFormDefault="unqualified">
 <import namespace="urn:mpeg:mpeg21:cel:core:2012" schemaLocation="cel-
core.xsd"/>
 <!-- === Acts === -->
 <element name="review" type="cel-core:Action" substitutionGroup="cel-
core:action"/>
<element name="archive" type="cel-core:Action" substitutionGroup="cel-
core:action"/>
 <!-- === Constraints === -->
 <element name="ipAccess" type="cel-core:IPAccessType" substitutionGroup="cel-
core:fact"/>
</schema>
```